Title: **Mission Plan Recognition: Developing Smart Automated Opposing Forces for Battlefield Simulations and Intelligence Analyses**

Suggested Topics: **Modeling and Simulation, Cognitive and Social Issues**

Authors:

Georgiy M. Levchuk
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x267
Fax: 781-935-4385
e-mail: georgiy@aptima.com

Darby Grande
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x225
Fax: 781-935-4385
e-mail: dgrande@aptima.com

Krishna R. Pattipati
Professor, ECE Dept., UCONN
Storrs, CT
Phone: 860-486-2890
Fax: 860-486-5585
e-mail: krishna@engr.uconn.edu

Yuri Levchuk
Aptima Inc.,
1726 M Street, N.W., Suite 900
Washington, DC 20036
Phone: (202) 842-1548x323
Fax: (202) 842-2630
e-mail: levchuk@aptima.com

Alexander Kott
DARPA / IXO
3701 Fairfax Drive
Arlington, VA 22203-1714
Phone: (571) 218-4649
e-mail: Alexander.Kott@darpa.mil

Correspondence:

Georgiy M. Levchuk
Aptima Inc.,
12 Gill Street, Suite 1400
Woburn, MA 01801
Phone: 781-935-3966x267
Fax: 781-935-4385
e-mail: georgiy@aptima.com

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUN 2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**
4. TITLE AND SUBTITLE **Mission Plan Recognition: Developing Smart Automated Opposing Forces for Battlefield Simulations and Intelligence Analyses**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **DARPA / IXO,3701 Fairfax Drive,Arlington,VA,22203-1714**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
13th International Command and Control Research and Technology Symposia (ICCRTS 2008), 17-19 Jun 2008, Seattle, WA

14. ABSTRACT
A key challenge for battlefield simulation is the estimation of enemy courses of action (COAs). Current adversarial COA development is a manual time-consuming process prone to errors due to limited knowledge about the adversary and its ability to adapt. Development of decision aids that can predict adversary?s intent and range of possible behaviors, as well as automation of such technologies within battlefield simulations, would greatly enhance the efficacy of training and mission rehearsal solutions. In this paper, we describe the development of OPFOR agents that can intelligently learn BLUEFOR?s mission plan. This knowledge will allow OPFOR agent to reason about the intent of BLUE and counteract accordingly to prevent/influence the future BLUEFOR?s operations by affecting current operations, challenging BLUE?s resources, and preparing OPFOR for future battles.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**41**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Abstract

A key challenge for battlefield simulation is the estimation of enemy courses of action (COAs). Current adversarial COA development is a manual time-consuming process prone to errors due to limited knowledge about the adversary and its ability to adapt. Development of decision aids that can predict adversary’s intent and range of possible behaviors, as well as automation of such technologies within battlefield simulations, would greatly enhance the efficacy of training and mission rehearsal solutions.

In this paper, we describe the development of OPFOR agents that can intelligently learn BLUEFOR’s mission plan. This knowledge will allow OPFOR agent to reason about the intent of BLUE and counteract accordingly to prevent/influence the future BLUEFOR’s operations by affecting current operations, challenging BLUE’s resources, and preparing OPFOR for future battles.

1. Motivation: Modeling Adaptive Opposing Forces

A key challenge for battlefield simulation is the estimation of enemy courses of action (COAs). Current adversarial COA development is a manual time-consuming process prone to errors due to limited knowledge about the adversary and its ability to adapt. Development of decision aids that can predict adversary’s intent and range of possible behaviors, as well as automation of such technologies within battlefield simulations, would greatly enhance the efficacy of training and mission rehearsal solutions.

Under the sponsorship from DARPA, Aptima Inc. has conducted an SBIR Phase I project to develop the Automated Collateral Tactics for OPFOR Responses (ACTOR) module for battlefield simulations. This module will allow automating control of opposing force’s (OPFOR) simulated units while requiring minimal input from the simulation operators based on the mission environment and the commander’s objectives. ACTOR modeling is based on dynamically learning the command and control (C2) behavior patterns of BLUE forces (BLUEFOR) and consequently generating adaptive OPFOR plans aimed at achieving the highest degree of deception and system-wide effects on the BLUEFOR C2 team. The ultimate goal of the ACTOR framework is to enable the design of advanced collaborative planning tools to support development of BLUEFOR courses of action.

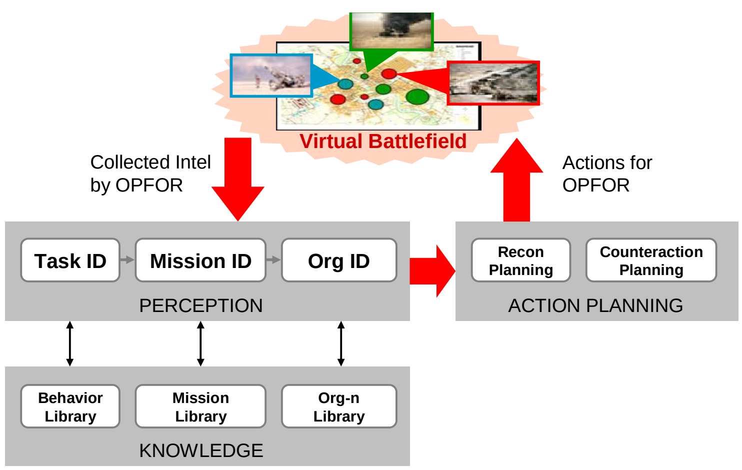


Figure 1: ACTOR Technology Components

ACTOR technology consists of the three components defining automated intelligent adversary (Figure 1):

- **Knowledge component** contains libraries of BLUEFOR actions signatures, mission patterns, and organizational structures and OPFOR action impact / BLUEFOR responses learned over time by ACTOR from interactive plays between various OPFOR and BLUEFOR teams;
- **Perception component** contains parametric inference algorithms enabling OPFOR’s identification of current and future actions, missions, and structures of BLUEFOR;

- **Action planning component** contains models for developing OPFOR’s reconnaissance plans and plans of (counter)actions against BLUE forces.

ACTOR will create OPFOR agents that can intelligently learn what BLUEFOR are doing and adapt their behaviors. This technology can bring several benefits to current Command and Control processes. First, ACTOR will enhance Intelligent Preparation of the Battlefield (IPB) process and Courses of Actions (CoAs) planning by allowing more accurate predictions of adaptive adversaries in today’s and tomorrow’s complex and asymmetric environments. Second, ACTOR will enable faster and more efficient training of US force commanders and staff against the simulated forces that mimic the adaptability of current enemies. And finally, ACTOR will allow quick mission rehearsal and collaborative wargaming to improve the readiness of BLUE forces.

In this paper, we describe one of the inference algorithms in perception component of ACTOR model that performs the recognition of BLUEFOR’s mission plan. The knowledge of BLUEFOR’s mission will allow OPFOR agent to reason about the intent of BLUE and counteract accordingly to prevent/influence the future BLUEFOR’s operations by affecting current operations, challenging BLUE’s resources, and preparing OPFOR for future battles.

The paper is organized as follows. In Section 2 we summarize current research in plan and behavior recognition. Section 3 describes our probabilistic plan identification algorithm. We present the use case and simulation analysis results in Section 4 and provide conclusions and future research directions in Section 5.

2. Related Research in Plan Recognition

Plan recognition is the process of inferring another side’s plans or behaviors based on observations of its interaction with the environment. Several applications of plan recognition have been developed in the last decade. Most of the automated plan recognition models, however, have severe limitations to be used by OPFOR agents:

- **Traditional utility-based plan recognition** infers the preferences of the actors and selects the plan that achieves the highest static or expected utility. Maximum-utility plan recognition models (Mao and Gratch, 2004; Blythe, 1999) cannot track the plan evolution over time as the utility of action execution mostly does not change while the actions in the plan are executed.
- **Traditional probabilistic plan tracking and actor profiling** looks at patterns of activities performed by a single individual or the whole group to determine its role, threat indicator, intent, goal, or future actions. This approach *does not allow tracking of coordinated and interdependent actions by multiple actors* in both space and time. For example, criminal clustering models (Stolfo et al., 2003) have only dealt with single relationship-based group identification, while spatial criminal forecasting (Brown, Dalton, and Hoyle, 2004) have relied only on the demographic information, areas of concentration of adversarial actors, and the locations of hostile events. Statistical temporal event analysis techniques, such as Hidden Markov Models (Schrodt and Gerner, 2001; Singh et al., 2004), Bayesian Networks (Tu et al., 2006), Markov Decision models (Yin et a., 2004), decision tree-based models (Avrahami-Zilberbrand, and Kaminka, 2005), and conditional hierarchical plans (Geib and Harp, 2004) can reliably forecast behavior of only single actor, dyadic relationships, or a group. This behavior representation assumes that only a single action can happen at any time. Each single actor or group and its actions may look benign, but only by analyzing combined interactions can one discern the true nature of behavior and enable early predictions of future hostile activities.
- **Traditional interactions analysis models** – including differential equations (Turchin, 2003), interaction-events data analysis (Gerner et al., 2002; O’Brien, 2004), game-theoretic models (Brams and Kilgour, 1988), agent-based simulations (Popp et al., 2006), and others – need to be pre-populated with a large amount of data. A significant amount of noise events (text parsing errors, misclassifications, missed information, and deceptions) contribute to misleading forecasts (false alarms

and false positives – the recognition of potential threats that have little or no impact) due to the sensitivity of these models to input parameters. In addition, different models work at different levels of granularity, with no common analytical and software framework developed to integrate model inputs and outputs (Popp et al., 2006). Very few of these models were able to "remove the noise" from the input data, and none of the models were able to work with data sources at different levels of granularity.

Instead of single actor plan recognition, the OPFOR needs to learn the mission of BLUEFOR that consists of multiple units and performs coordinated activities constrained by the BLUE's organizational structure. Therefore, we need to account for the resource and organizational constraints of BLUE forces, the utility and probabilistic nature of the actions, the uncertainty in dynamic observations about BLUE's activities, and the fact that many activities might happen in parallel. In spirit, our models of perception for OPFOR agent are close to team plan recognition research (Kaminka, and Pynadath, 2002). Our approach differs in the quantitative representation of the organizational structure and mission plans, and the algorithms that we use for discovering the hidden organization and mission of BLUEFOR from noisy observations.

3. Method: Probabilistic Mission Plan Identification

ACTOR mission recognition algorithms are based on hypothesis testing principles: the algorithm selects the mission(s) from the hypotheses set that **best explains** the OPFOR's observations (Figure 2). Each mission is matched against the set of observations to determine its most likely state. The match between the mission and its state is scored with the a-posteriori function, and then the likelihood function is used to rank-order the missions from the hypotheses set.

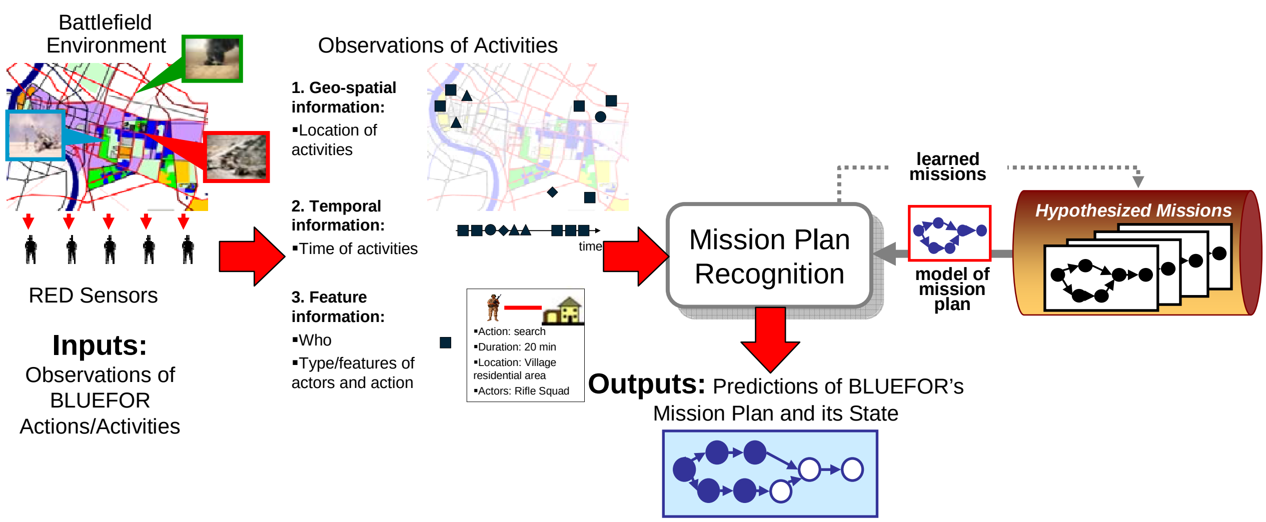


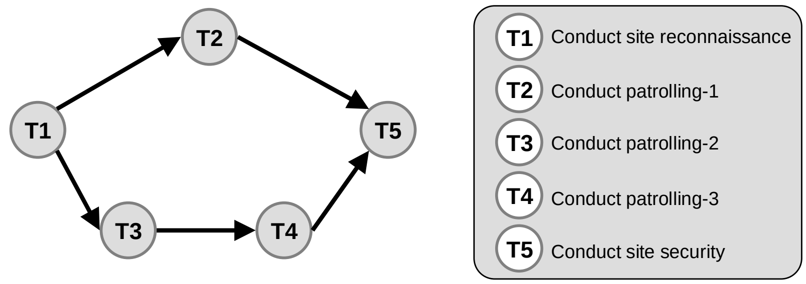
Figure 2: Mission Plan Recognition Process

The set of observations is obtained by OPFOR sensors, which include its operatives, insurgents, fighters, civilians and local informants. The observations are noisy events about the operations that individual BLUE force units are conducting (e.g., patrols, searches, security operations, force march, checkpoint setup, etc.). These events contain three information elements (Figure 2): (i) geo-spatial information – indicating the location of activities; (ii) temporal information – indicating the time of activities; and (iii) feature information – indicating the type of actions, participating units, resources used, etc.

3.1. Mission Plan and Military Assets Representation

Formally, a *mission* is defined as a plan that BLUEFOR has created and going to follow. We follow a formal planning model defined in (Levchuk et al., 2002). Missions consist of *tasks* that individual BLUEFOR units will execute. In order to define the mission, we need to specify its structure, the set of tasks, and the resources required to execute theses tasks. The *mission structure* (Figure 3) is defined as a

directed acyclic graph $G=(V,E)$ termed *task graph* (Levchuk et al., 2002), where the set of graph nodes $V=\{T_1,T_2,...,T_N\}$ represents the tasks of the mission and a set of directed edges $E=\{e_{ij}=<T_i,T_j>\}$ represents the precedence constraints among tasks (so that tasks can be started only after all their predecessor tasks are completed). We use this formulation for the reasons of simplicity, but it can be extended to represent task networks with conditional nodes and temporal constraints (Vidal and Bidot, 2001; Rossi, Venable, and Yorke-Smith, 2003).



(a) Example of Mission Task Graph (b) Example of Mission Task List

Figure 3: Example of Mission Plan

Type	Description
SCR	Secure (areas, sites);
EN	Envelope-Isolate-engage (of the enemy forces)
TRSP	Transport/MED Evacuation (of forces, soldiers, civilians, etc.)
MAN	Manage-Maintain-Setup (checkpoints, facilities, buildings)
REC	Ground Recon-Search – to search buildings, routs, collect intelligence on the ground
INT	Interrogate civilians, criminals, enemy combatants
FIRE	Non-precision Fire against enemy – including direct and indirect fire capability, such as missiles, bombs, artillery, etc.
DTN	Detain enemy combatants, civilians, etc.
PTRL	Patrol/Force presence ops – patrolling operations, which very often are conducted to enforce the curfews, show presence and discourage criminals from illegal actions, and militia from attacks

Figure 4: List of Resource Types

To model the task execution and allocation of resources, we define the list of resource types (Figure 4). Then, for each task T_i, we define the *resource requirement vector* $[R_{i1},R_{i2},...,R_{iL}]$ (Figure 5), and for each unit/asset P_m in the BLUE’s organization we define *resource capabilities vector* $[r_{m1},r_{m2},...,r_{mL}]$ (Figure 6). Here, R_{il} is the number of units of resource *l* required for successful processing of task T_i and r_{ml} is the number of units of resource type *l* available on platform P_m ($l=1,...,L$, where *L* is the number of resource types). The task T_i execution is successful if the vector of applied resources from the BLUE’s assets is component-wise more or equal to the task’s resource requirements: $\sum_{m=1}^{K} r_{ml}\cdot w_{im}\geq R_{il},\quad i=1,..,N; l=1,..,L;$ where $w_{im}=1$ if asset P_m is assigned to task T_i. The application of asset’s resource to the task can be viewed as an individual action by this asset, and can thus be observed by OPFOR’s sensors.

Tasks	SCR	EN	TRSP	MAN	REC	INT	FIRE	DTN	PTRL

Site reconnaissance	0	0	0	0	2	0	0	0	0
Building search	2	0	0	0	1	0	0	0	0
Checkpoint setup	1	0	0	1	0	0	0	0	0
Attack OPFOR positions	0	2	0	0	0	0	4	0	0
Site security	3	0	0	0	0	0	0	0	0
Resupply ops	1	0	3	0	0	0	0	0	0
Detain OPFOR members	0	0	0	0	0	2	0	2	0
Patrolling	0	0	0	0	0	0	0	0	3

Figure 5: Example of Task Resource Requirements

BLUE Units	SCR	EN	TRSP	MAN	REC	INT	FIRE	DTN	PTRL
RFL Squad	1	0	0	1	0	0	0	1	1
Tank	0	1	0	0	0	0	1	0	0
Rec Squad	1	0	0	0	1	0	0	0	0
MP	0	0	0	1	0	1	0	1	1
Humwee	0	0	1	0	0	0	0	0	0

Figure 6: Example of Asset Resource Capabilities

3.2. Mission State Representation

The state of mission $G=(V,E)$ is then defined as a labeled network $\Omega=(V,E,S)$, where $S=\{s_i^T \in \{0,1\} \mid i=1,...,N\}$ is a set of task states that correspond to node labels ($s_i^T=1$ if the task T_i is completed; otherwise $s_i^T=0$). Figure 7 shows the example of the mission and its state.

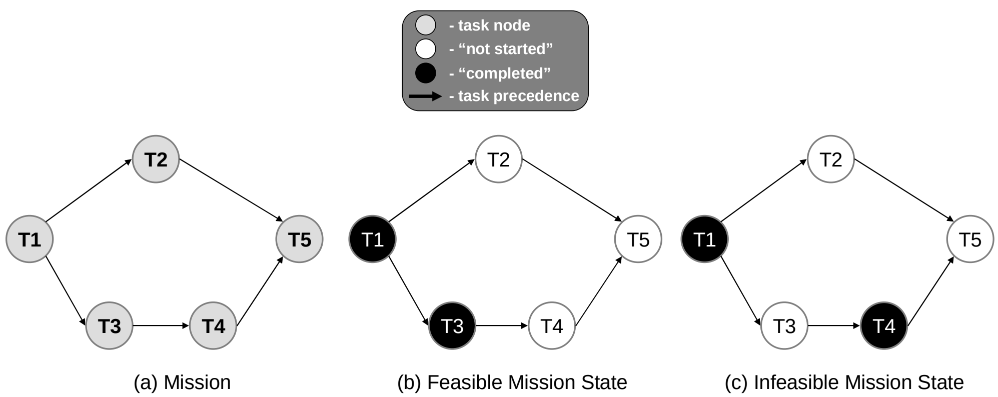


Figure 7: Example of Mission and its State

Mission states can be feasible and infeasible. A *feasible* mission state (Figure 7b) represents task execution satisfying precedence constraints (that is, if $s_j^T=1$ then $s_i^T=1, \forall i: e_{ij}\in E$ - i.e., all parents of the completed task are themselves completed). An *infeasible* state is such that this condition is not satisfied. For example, in Figure 6c the task T4 is indicated as “completed”, while its predecessor T3 is not.

3.3. Knowledge and Data Available to OPFOR

The OPFOR agent has a *knowledge* of a set of feasible missions that BLUE may conduct. This knowledge could have been generated from experience of the OPFOR’s battles against BLUEFOR, studying the BLUE’s doctrine, etc. The models to build this knowledge are outside the scope of this paper.

Figure 8 shows an example of the set of feasible missions that BLUEFOR can conduct; this set is assumed known by OPFOR. In Figure 8, we color-coded the tasks of the same type that occur in different locations. As the result, we capture the spatial information (information about task locations), temporal information

(sequencing of tasks according to precedence in the mission), and type information (overlap in the types of activities that need to be performed). As an example, missions M1 (Recon and patrolling) and M2 (ground stability and defensive ops) have two common tasks, (“site recon” and “site security”), and are qualitatively and quantitatively distinguished by other mission-unique tasks.

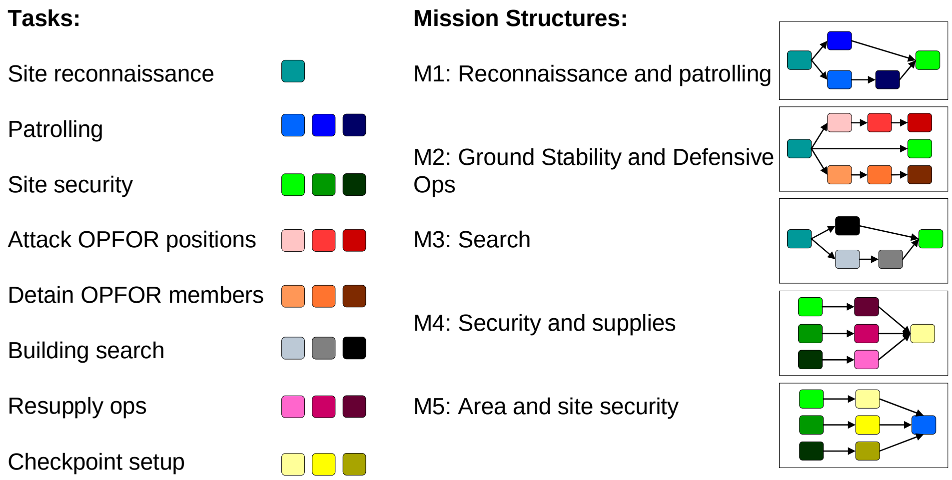
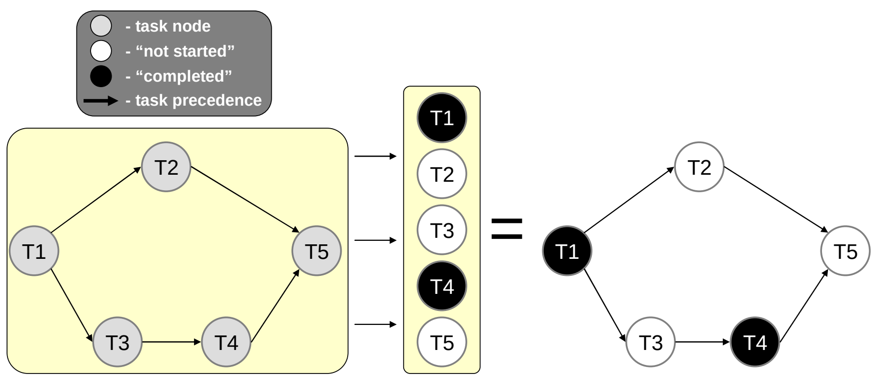


Figure 8: The Set of BLUE Missions for the Experiment



(a) Mission (b) Observations (c) Observed Mission State

Figure 9: Example of Observed Mission State

Due to the fact that OPFOR agent does not know the true BLUEFOR’s mission or its state, we can say that the mission state is *hidden*. From the observations about tasks’ states, for each mission plan in the hypothesis set the OPFOR agent composes *observed mission state* (Figure 9). For a specific mission plan, the observed mission state may be infeasible due to missing data, errors in task identification, deceptions and irrelevant observations. In order to identify what mission plan is in progress and what is its current state, we need to find mission plans and their states that not only provide best match to the observed data, but also are feasible given the resources of the organization.

3.4. Probabilistic Inference Model: Mission Plan Recognition

<u>The Mission State Influence Model.</u>

Relationship between actual mission state and observations can be represented by the following model. We view the task as a random variable s_i^T that can take values $s_i^T = s \in \{0,1\}$. Each such random variable (task) is not directly observable, and is assumed to have prior probability of being completed equal to

$$p_i(s) = p\{s_i^T = s\} \qquad [1]$$

Because task execution depends on other tasks and on availability of resources, we define the dependencies between random variables s_i^T in the form of the *conditional Markov field*, that is a conditional probability of a state of a task given the state of its neighbors:

$$p_i(s \mid s_N) = p\{s_i^T = s \mid s_j^T = s_j, e_{ji} \in E\}, s_N = \{s_j, e_{ji} \in E\} \qquad [2]$$

where $s_N = \{s_j, e_{ji} \in E\}$ is a set of states of predecessor tasks for task T_i. In [2] we intentionally included only the predecessors of the task T_i, because this probability can be tied to the time task T_i *can* be executed (i.e., when all predecessors have been completed). Note that the Markov property means that $p\{s_i^T = s \mid s_j^T = s_j, \forall j\} = p\{s_i^T = s \mid s_j^T = s_j, e_{ji} \in E\}$. Obviously, probability of a task to have a state 1 (“completed”) is zero if at least one of the predecessor tasks has not been completed:

$p_i(s = 1 \mid s_N : \{0\} \in s_N) = 0$, $p_i(s = 0 \mid s_N : \{0\} \in s_N) = 1$

We then define a probability of task completion when all predecessors have been completed:

$$p_i(s_i = 1 \mid s_{N\{i\}} = 1) = p_i^c \qquad [3]$$

The Markov assumption can be used to define the joint mission state (prior) probability as:

$$p(S) = p\{s_i^T = s_i, i = 1,\dots, N\} = \prod_i p\{s_i \mid s_{N(i)}\} \qquad [4]$$

<u>The Mission State Observation Model.</u>

The task can emit the observation about its state – a random variable $o_i^T \in \{0,1\}$. The observation process is defined using conditional probability

$$p_i(o \mid s) = p\{o_i^T = o \mid s_i^T = s\} \qquad [5]$$

This probability can be captured based on knowledge of the accuracy and availability of data collection resources and the model of the BLUEFOR’s deception. We then define the observation probabilities as:

Probability of miss:

$$p_i(o_i = 0 \mid s_i = 1) = p_i^m \qquad [6]$$

Probability of false alarm:

$$p_i(o_i = 1 \mid s_i = 0) = p_i^f \qquad [7]$$

Our setup of the observation model is equivalent to an assumption that the observations about mission state are modeled as a random field that is conditionally independent, that is:

$$p(O \mid S) = p\{o_i^T = o_i, i = 1,\dots, N \mid s_i^T = s_i, i = 1,\dots, N\} = \prod_{i=1}^{N} p_i(o_i \mid s_i) \qquad [8]$$

The Inference Model.

We identify two problems of mission plan identification:

Problem 1 (Mission State Identification): For a given mission plan (hypothesis) m, find its state $\hat{S}_m$ that most likely has generated the observations O. One of the possibilities is to use the maximum a-posteriori estimator:

$$\hat{S}_m = \arg\max_S p(S \mid O, m) \qquad [9]$$

Note that conditioning on mission m has been disregarded in the previous section for simplicity.

The solution in [9] is taken over the set of mission states that are feasible given current observations. This means that for tasks indicated as “completed” in the state definition, for which the observation about their execution was not available, we must find the asset assignment which does not violate assignment of other tasks.

Problem 2 (Mission Identification): For a given set of mission model plans M find the mission $\hat{m} \in M$ that has most likely generated the observations O. One example of the objective function is to use the maximum likelihood estimator:

$$\hat{m} = \arg\max_m p(O \mid m) \qquad [10]$$

3.5. Solution

To obtain an estimate of the mission’s state, i.e. a solution to [9], we first note that using [4] and [8] and the fact that $p(S \mid O, m) = \dfrac{p(O \mid S, m)\, p(S \mid m)}{p(O \mid m)}$ we get:

$$p(S \mid O, m) = \frac{1}{Z} \prod_i p_i(o_i \mid s_i)\, p_i(s_i \mid s_{N\{i\}}) \qquad [11]$$

where $z = \sum_S \prod_i p_i(o_i \mid s_i) \prod_i p_i(s_i \mid s_{N\{i\}})$

Then, the log-posterior of mission state is:

$$\log p(S \mid O, m) = \sum_i \log p_i(o_i \mid s_i) + \sum_i \log p_i(s_i \mid s_{N\{i\}}) + const \qquad [12]$$

Due to setup of mission precedence graph, the second component will only contain a reference to the predecessors that all have state “completed”, that is it will contain $\log p_i^c$ and $\log\left(1 - p_i^c\right)$. All other components are = 1, and hence $\log p_i(s_i \mid s_{N\{i\}}) = 0$. All we need to make sure is that the mission state is feasible – that is, we only consider $s_i = 1$ if $s_j = 1,\ j \in N\{i\}$.

As the result, we will have for a feasible mission state S:

$$\log p(S \mid O, m) \propto \sum_i \begin{matrix} 1_{\{s_i=1\}}\left[\log p_i^c + 1_{\{o_i=0\}} \log p_i^m + 1_{\{o_i=1\}} \log(1 - p_i^m)\right] \\ + 1_{\{s_i=0\}}\left[1_{\{S_{N\{i\}}=1\}} \log(1 - p_i^c) + 1_{\{o_i=0\}} \log(1 - p_i^f) + 1_{\{o_i=1\}} \log p_i^f\right] \end{matrix} \qquad [13]$$

There are several algorithms to find a solution that maximizes the expression in [13], which include:

- **Enumeration:** generate the set of all feasible states of the mission, score each state according to [14], and pick the maximum;
- **Greedy search:** iteratively update the state of a single mission’s task based on largest improvement in the objective function [13];
- **Stochastic search:** update the mission state iteratively (e.g., choose the task to update based on soft-max principle) and select a solution based on some strategy (e.g., using simulated annealing, which accepts the mission state with probability $p(S_{n+1}) = \begin{cases} 1, & f_{n+1} \geq f_n \\ \exp \frac{f_{n+1} - f_n}{T}, & otherwise \end{cases}$, where $f_n = \log p(S_n | O, m)$ and T is a decaying temperature).
- **AO* algorithm:** search through the state space of mission states and update the utility of task state changes during the backtracking.

To find the estimate of BLUEFOR’s mission plan, i.e. a solution to [10], we note that it is decomposed as:

$$p(O | m) = \sum_S p(O | S, m) p(S | m) = \sum_S \prod_i p_i(o_i | s_i) p_i(s_i | s_{N\{i\}}) \quad [14]$$

The summation in [14] is over all possible mission states, and in case the task graph complexity is high (i.e., the number of tasks is large and the precedence constraints are sparse), we find an approximation to this function using importance sampling algorithm (Srinivasan, 2002). For the purposes of our simulation results (see next section), we have used the full enumeration since the graph complexity in our study example was small.

4. Results: Assessing the Accuracy and Sensitivity of Mission Plan Recognition Algorithms

4.1. ACTOR Prototype Components

To assess the accuracy of developed predictive models, we have developed a prototype incorporating virtual C2 simulation and perception algorithms. The high-level architecture of ACTOR prototype is shown in Figure 10. The core of the ACTOR prototype solution is a Simulation Controller implementing a virtual battlefield C2 task. The controller was based on the distributed agent-based models of simulated asset control, that were able to implement three main actions: (i) move in the environment by routs and zones; (ii) engage other assets (simulation entities) in the environment – e.g., BLUEFOR assets can attack other assets; and (iii) sense information in the environment – e.g., detect enemy assets, classify assets, observe actions, etc. The assets were controlled to move and engage other assets in the environment based on the mission plan selected for the corresponding force. In our simulations we have assigned a single mission to BLUEFOR. The OPFOR assets were kept stationary in the environment and used only to provide sensed data feeds (detected entities and events) for predictions about BLUEFOR. The planning components developed by ACTOR in the future will fully automate the OPFOR assets.

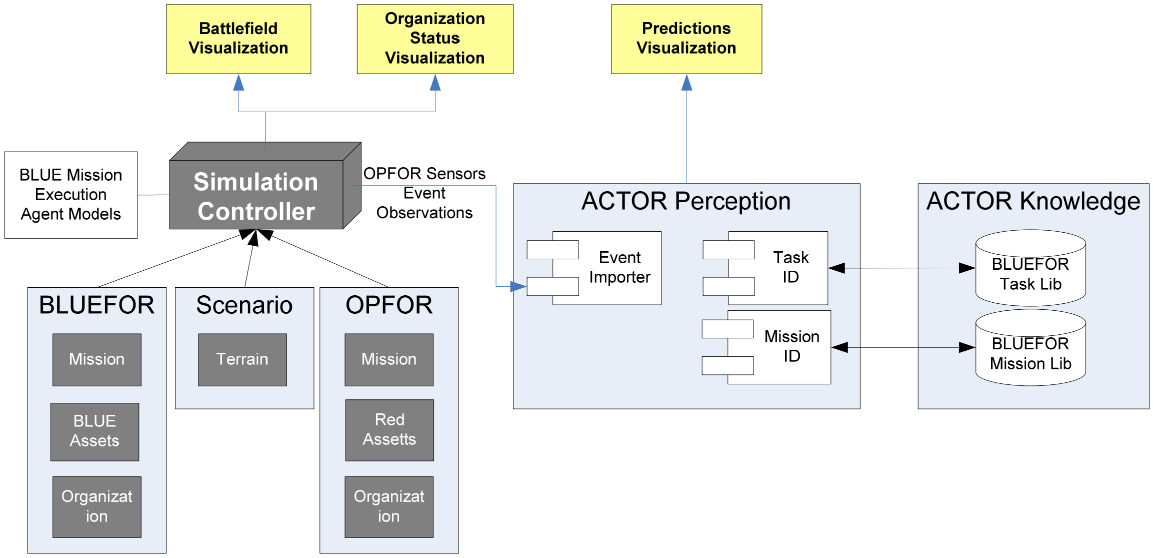


Figure 10: ACTOR Prototype --- Architecture

Three visualizations / User Interfaces have been implemented in ACTOR prototype: (a) battlefield visualization – which allowed to see the dynamic battlefield state, travels and engagements of opposing forces' assets (Figure 11); (b) organization status visualization – which showed the status of assets and commanders of BLUEFOR and OPFOR (e.g., task schedule, number of sensed targets, available resources, etc.); and (c) prediction visualization – which showed the likelihood scored of hypothesized BLUEFOR mission plans and the status of the plan over time.

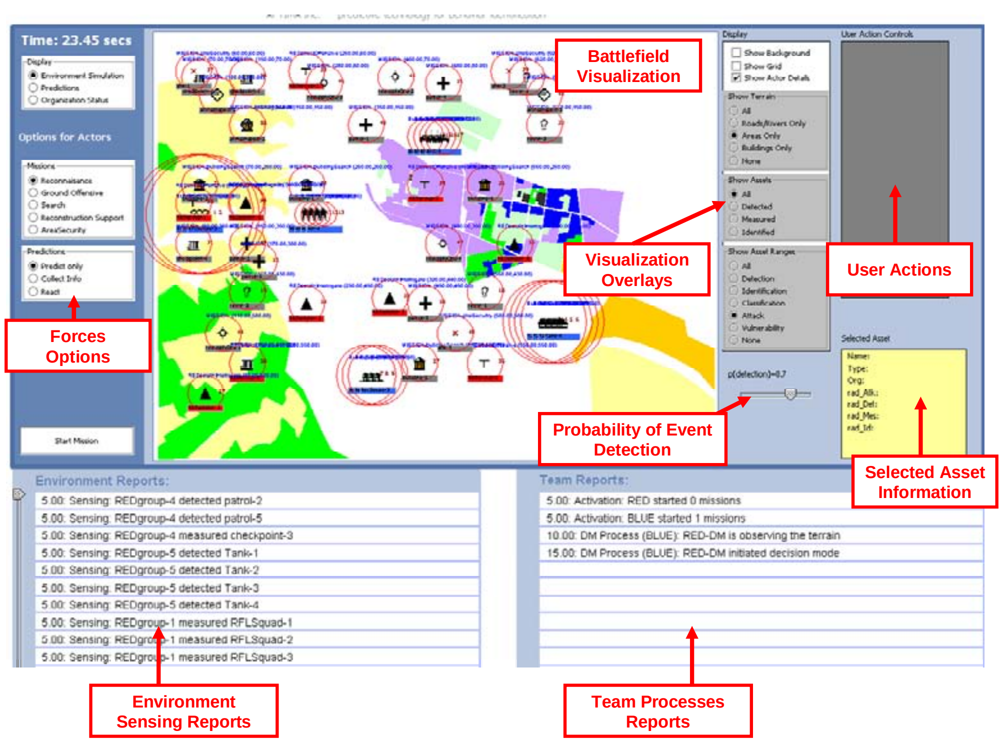


Figure 11: ACTOR Prototype --- Battlefield Visualization

ACTOR perception component was implemented in the prototype as a BLUEFOR’s mission plan identification, and was based on hypotheses testing principles. The knowledge of the hypotheses was stored in the model and retrieved at the time when prediction algorithm was evoked.

4.2. The Scenario Story

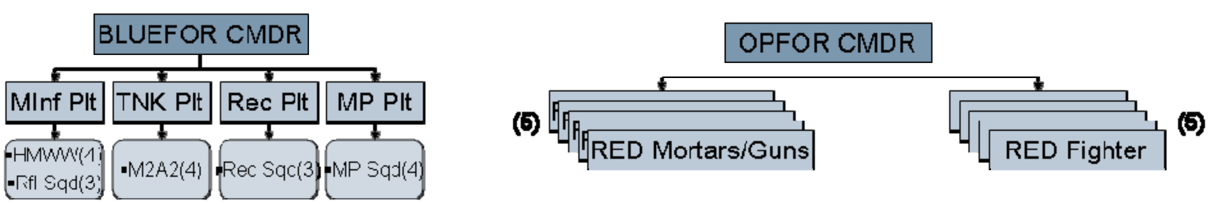
To evaluate a sensitivity of ACTOR predictions, we have created a synthetic scenario based on the following story. The battlefield was an urban terrain of a 3-rd-world country in which U.S. forces are conducting the stability operations and support missions. Continuous fighting with local militia has been undermining U.S. efforts in the region to support local government establish rule of law and provide for the population. The U.S. forces in the region designated the company-size units to conduct small-scale short-time operations, including (see Section 2.3 for complete definition of the following missions):

- Reconnaissance and patrolling
- Ground Stability and Defensive Ops
- Search
- Security and supplies
- Area and site security

The BLUE force organization structure (see Figure 12(a)), headed by a Chief Warrant Officer and a captain or major, consisted of 4 platoons (25-60 people each; headed by warrant officers and first or second lieutenants):

- Mechanized infantry platoon
  - equipped with four Humwees
  - organized with a platoon headquarters and three rifle squads
  - the platoon leader and his headquarters mounted in one Humwee, and the squads mounted in the other three
- Tank platoon
  - four main battle tanks organized into two sections, with two tanks in each section
  - the platoon leader (Tank 1) and platoon sergeant (Tank 4) are the section leaders. Tank 2 is the wingman in the platoon leader's section, and Tank 3 is the wingman in the platoon sergeant's section
- Reconnaissance platoon
  - 1 officer and 18 enlisted soldiers
  - organized into a platoon headquarters and three squads
  - equipped with individual weapons, night vision devices, and communications equipment
  - There are a total of 16 M16A2 rifles and 3 M203 grenade launchers (one per squad)
- Military Police platoon
  - one officer and 29 enlisted soldiers
  - a headquarters element and four military police squads

The OPFOR consisted of 5 cells and cell leaders. Each cell possessed mortar and heavy guns and was capable of attacking BLUEFOR with IEDs/RPGs. The cell members mixed with local population and conducted reconnaissance activities to learn about BLUEFOR.



(a) BLUEFOR Organization (b) OPFOR Organization

Figure 12: Use Case --- OPFOR and BLUEFOR Military Organizations

The data collected by OPFOR entities consisted of observed *movements* of BLUEFOR units (Ex: “Rec Squad moved to Building A at 10:03”), observed *actions/engagements* of individual BLUEFOR units (Ex: “Rfl Squad manned site B at 12:54”), and *additional information* about BLUEFORE units and engagements (e.g., type of unit, type of action performed, etc.). It was then quantitatively represented as (i) geo-spatial information – indicating the location of activities; (ii) temporal information – indicating the time of activities; and (iii) feature information – indicating the type of actions, participating units, resources used, etc. This data was fed to ACTOR perception component for predictive inference to enable recognition over time which BLUEFOR mission is taking place and what its current state is – so that OPFOR can develop actions against BLUEFOR entities of highest impact to BLUE and lowest cost to RED.

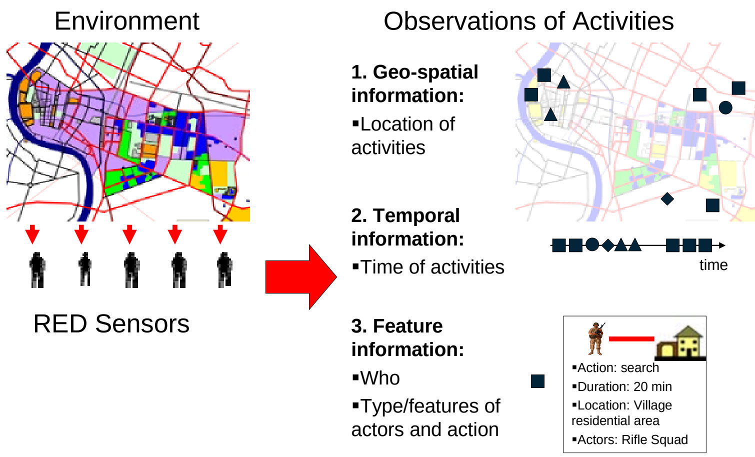


Figure 13: Use Case --- OPFOR Data Feeds

4.3. Sensitivity Results

We have conducted computational experiments with BLUEFOR performing all missions from the set in Figure 8 and evaluating the performance of the mission plan recognition algorithm for different levels of uncertainty.

First, we show how the ACTOR prediction works over time. In Figure 14, we show the predictions obtained by ACTOR technology at different sampling times from the start of the mission. As we can see, ACTOR first cannot recognize the mission correctly, because the observed data does not allow a good distinguishability from other mission. Over time, as BLUE continues performing its mission, ACTOR makes a correct inference about mission structure while at first not correctly identifying the state of the mission. With time, ACTOR predictions are improved.

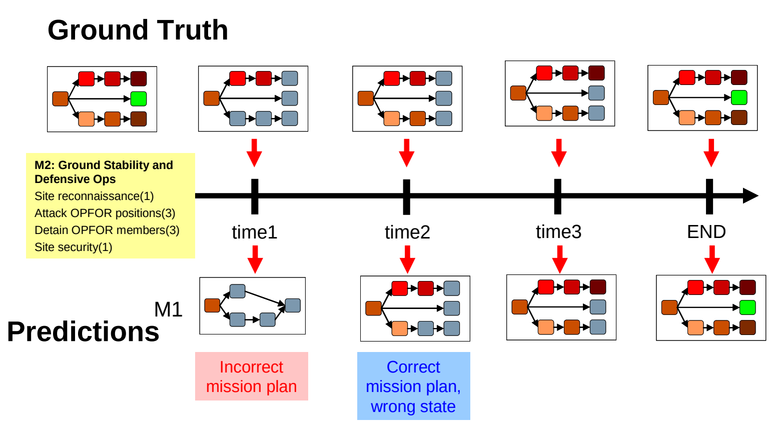
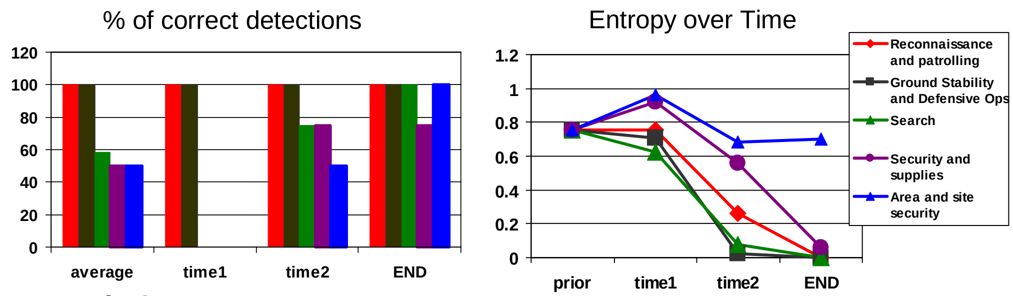


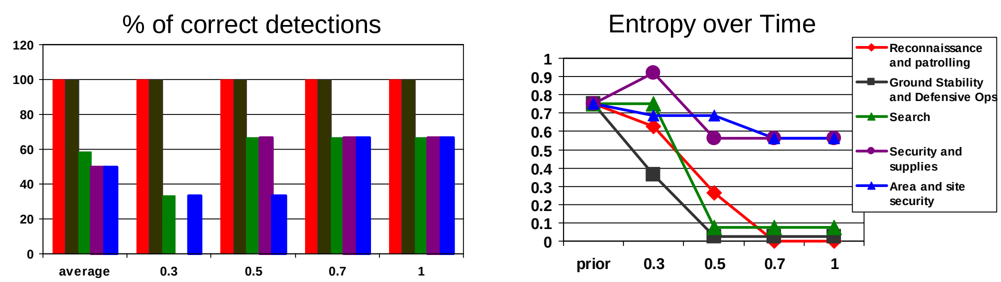
Figure 14: Simulation Results --- ACTOR Predictions over time



- **Conclusions:**
  - High detection accuracy (>=75%) in the middle of the mission
  - Detection accuracy data supported by entropy (power of estimator)
  - Detection improves towards the end of the mission as more actions are detected

Figure 15: Simulation Results --- effect of time on ACTOR Predictions

In order to assess performance of ACTOR algorithms, namely the sensitivity to the ground truth, sensitivity to probability of event detection, and sensitivity to amount of data ACTOR needs, we conducted simulation runs for all 5 types of missions (Figure 8), different levels of event detection probability and different sampling times (time when predictions are made). See results and conclusions in Figures 15-16.



- **Conclusions:**
  - Improved accuracy of event detection increases accuracy of predictions
  - Sensitivity to specific mission plan classes is observed

Figure 16: Simulation Results --- effect of event detection on ACTOR Predictions

As we can see, preliminary results suggest that ACTOR provides a high level of detection accuracy (>75%) early in the mission stage. The entropy results, which is inversely proportional to the power of the algorithm to provide the correct decisions “not accidentally”, supports the conclusion that ACTOR predictions are also robust.

5. Conclusions and Future Research

This paper described the novel models and algorithms to conduct mission plan recognition and presented the application of this technology to developing intelligent OPFOR agents. Such agents can be used in gaming simulations for human training and human-in-loop experiments, and can be enhanced to behave based on specific training or experimentation goals (e.g., train against the RED forces with most adaptability). In addition, ACTOR capabilities can be used operationally in conducting mission rehearsals and intelligence analysis tasks. The results shown in this paper suggest that ACTOR perception capability can achieve high accuracy of the mission plan identification under significant information gaps. Further research is needed to explore the solution envelope of this technology and compare against human-based RED decision making.

The algorithms described in this paper are based on parametric models and hypotheses testing principles. These parameters and hypotheses can be trained from the historic data. We are currently developing algorithms for supervised and unsupervised parameter learning from labeled and partially labeled datasets.

To be fully successful again opposing forces, predictive capabilities need to be combined with development of actions to collect the most critical information and actions to counteract the opposing forces. The effectiveness of such actions must be learned over time based on the experiences and interactions with opposing forces. The proposed solution can also be generalized to recognize more adaptive plans, where in addition to precedence constraints we model the event- and information-based mission changes. The design of information collection and disruption actions, model training, and adaptive plan recognition form the basis of our continued research in the area of automated synthetic agents.

References:

Avrahami-Zilberbrand, D., and G. Kaminka, “Fast and complete symbolic plan recognition”, *Proceedings of IJCAI*, 2005

Blythe, J., “Decision-Theoretic Planning”, *AI Magazine*, 20(2), 1999.

Brams, S. J., and Kilgour, D. M. (1988). *Game Theory and National Security*. New York: Basil Blackwell.

Brown, D.E., J. Dalton, and H. Hoyle, “Spatial forecast methods for terrorism events in urban environments”, *Proceedings of the Second Symposium on ISI*, 2004, pp. 426-435

Geib, C.W., and Harp, S., “Empirical analysis of a probabilistic task tracking algorithm”, *AAMAS workshop on Modeling Other agents from Observations*, (MOO-04), 2004.

Gerner, D.J., P.A. Schrodt, Ö. Yilmaz, and R. Abu-Jabr, “Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for a Post Cold War World”, *2002 Annual Meeting of the American Political Science Association*, 29 August - 1 September 2002

Kaminka, G., and Pynadath, D., “Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach”, *Journal of Artificial Intelligence Research*, 17 (2002), 83-135

Levchuk, G., Y. Levchuk, and K. Pattipati, “Identifying Command, Control and Communication Networks from Interactions and Activities Observations”, *Command and Control Research and Technology Symposium*, 2006, San Diego, CA.

Levchuk, G., Levchuk, Y., Luo, J., Pattipati, K. and Kleinman, D. “Normative Design of Organizations - Part I: Mission Planning”, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 32, No. 3, May 2002, pp. 346-359.

Mao, W., and Gratch, J., “Decision-theoretic Approach to Plan Recognition”, *ICT Technical Report*, ICT-TR-01-2004, available at http://ict.usc.edu/files/publications/ICT-TR-01-2004.pdf

O’Brien, S.P. 2004 “Near-Term Forecasts of Crisis and Instability Using Text-Based Events (NEAR-TERM FORECITE) Date: 04 OCT 2004”, *Proceedings of the 72nd MORS Symposium*, 22-24 June.

Popp, R., D. Allen and C. Cioffi-Revilla, “Utilizing Information and Social Science Technology to Understand and Counter the 21st Century Strategic Threat,” in *Emergent Information Technologies and Enabling Policies for Counter-Terrorism*, (Eds. R. Popp, J. Yen), Wiley-IEEE Press, pp. 1-27, Jun 2006.

Rossi, F., Venable, B., and Yorke-Smith, N. “Preferences and Uncertainty in Simple Temporal Networks”, *Proc. of the CP-03 Workshop on ”Handling Change and Uncertainty”*, Cork, Ireland, 2003.

Srinivasan, R. *Importance sampling - Applications in communications and detection*, Springer-Verlag, Berlin, 2002.

Vidal, T., and Bidot, J. “Dynamic Sequencing of Tasks in Simple Temporal Networks with Uncertainty”, *Proc. of the CP-01 Workshop on ”Constraints and Uncertainty”*, Paphos, Cyprus, 2001.

Stolfo, S.J., et al., “Behavior Profiling of Email,” *Proceedings of NSF/NIJ Symposium on Intelligence & Security Informatics*, 2003

Schrodt, P.A., and D.J. Gerner, (2001) Analyzing International Event Data: A Handbook of Computer-Based Techniques, *Draft Book*; available at http://web.ku.edu/keds/papers.dir/automated.html

Singh, S., Allanach, J., Tu, H., Pattipati, K.; Willett, P., “Stochastic Modeling of a Terrorist Event via the ASAM system”, *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, The Hague, The Netherlands, Oct. 2004

Tu, H., Allanach, J., Singh, S., Pattipati, K.R., and Willett, P., “Information Integration via Hierarchical and Hybrid Bayesian Networks”, *IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans*, Vol. 36, No. 1, 2006, pp. 19-33

Turchin, P., *Historical Dynamics: Why States Rise and Fall*, Princeton: Princeton University Press, 2003

Yin, M.-H., Ou, H.-J., Gu, W.-X., Lu, Y.-H., Liu, R.-X., “Fast probabilistic plan recognition without plan library”, *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, Vol. 1, Issue , 26-29 Aug. 2004, pp. 157 - 162

APTIMA®
HUMAN-CENTERED ENGINEERING

Mission Plan Recognition: Developing Smart Automated Opposing Forces for Battlefield Simulations and Intelligence Analyses

Georgiy Levchuk, PhD (Aptima)
Darby Grande, PhD (Aptima)
Krishna Pattipati, PhD (UCONN) Yuri Levchuk, PhD (Aptima)
Alex Kott, PhD (DARPA)

Presented at 2008 CCRTS

www.aptima.com
Woburn, MA ▪ Washington, DC

Overview

- The problem
- Solution concept
- Model description
- Use case and simulation results
- Future plans

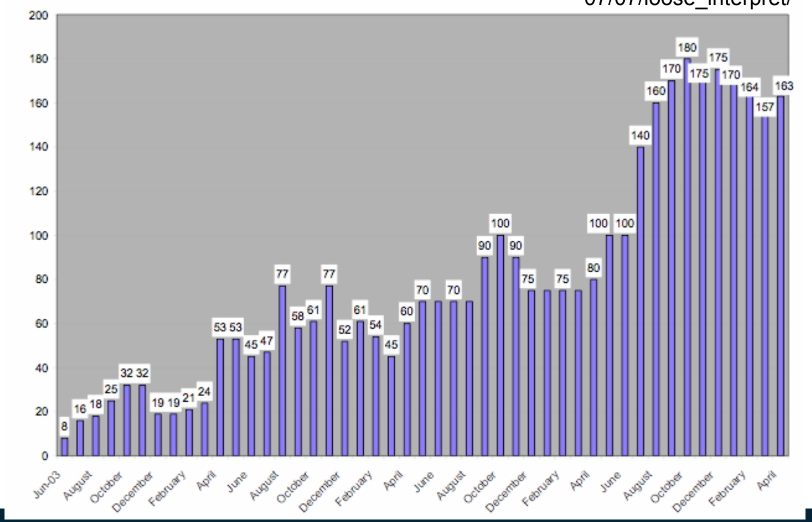
APTIMA®
HUMAN-CENTERED
ENGINEERING

The Problem

- Adversaries constantly adapt to actions of U.S. forces
  - Need to embed adaptation in predicting the adversarial behaviors
- Current training is standardized, slow, and mismatches the asymmetric requirements of current wars and conflicts
  - Current models of OPFOR in training simulations are hard to construct and are not adaptive

http://www.brianbeutler.com/2007/07/loose_interpret/

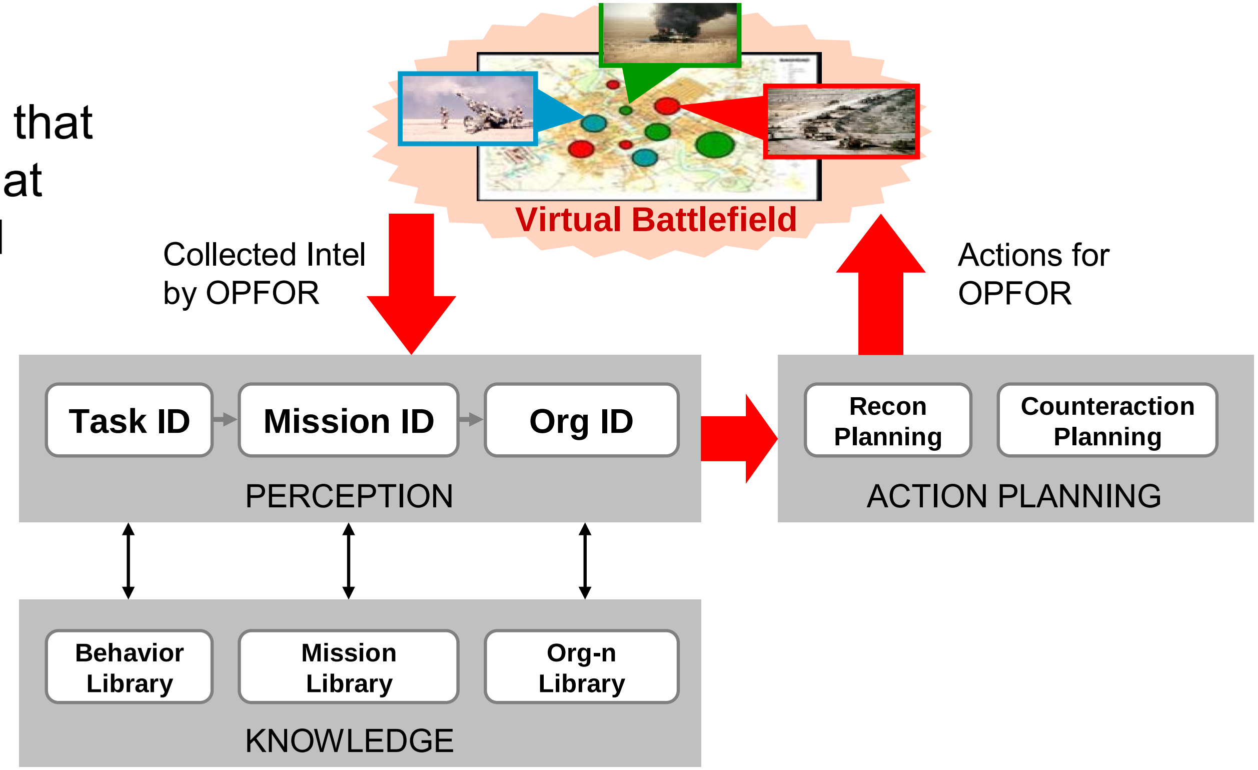
NUMBER OF DAILY ATTACKS BY INSURGENTS AND MILITIAS[7]



Proposed Solution

Solution

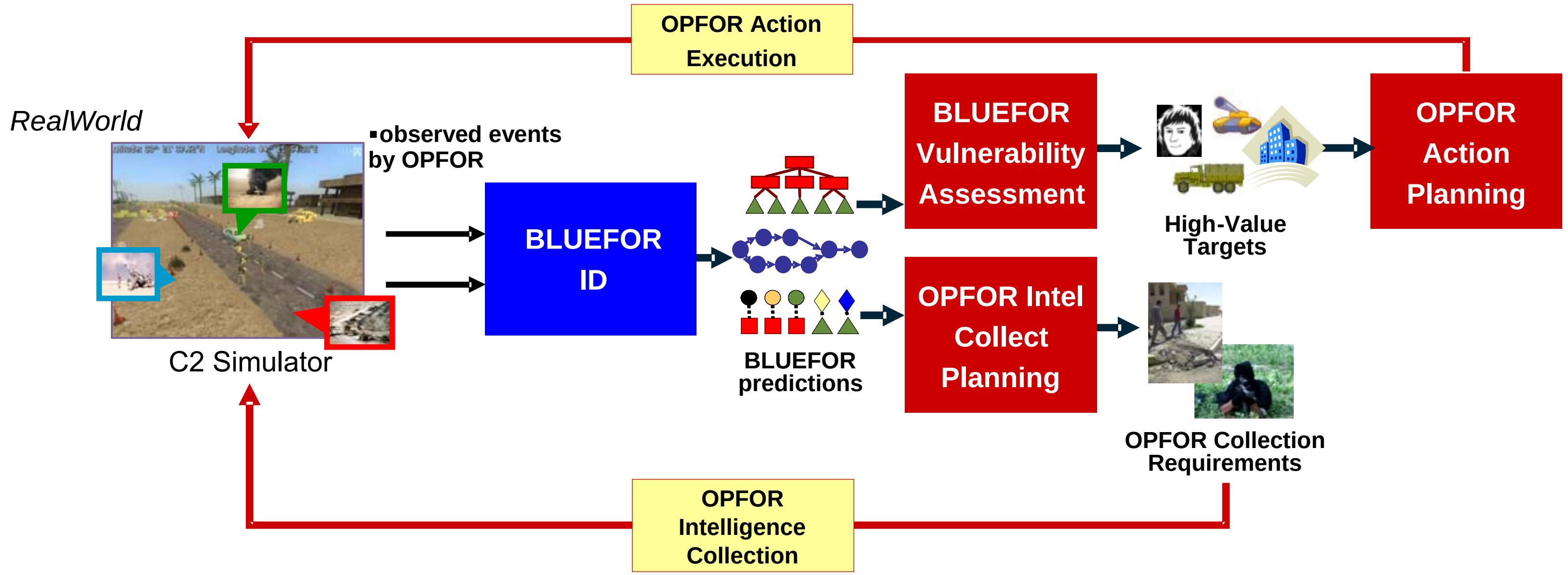
- Develop ***OPFOR agents*** that can intelligently **learn** what BLUEFOR are doing and **adapt their behaviors**



Benefits

- Better **prediction** of adaptive enemy
- Faster and more efficient **training**
- Collaborative war-gaming/**mission rehearsal**
- Use of same models for developing **BLUE decision alternatives**

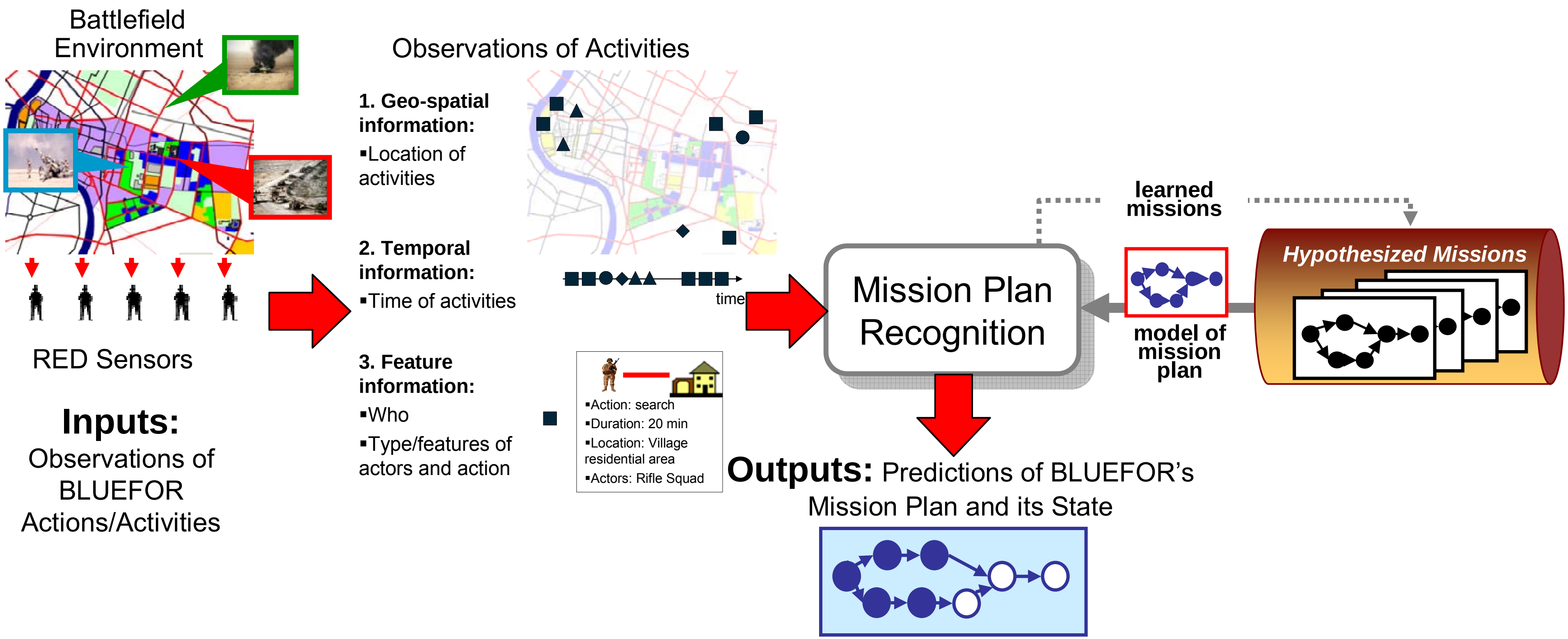
The “Big Picture”



- **Phase I effort:**
  - Models for all components
  - BLUEFOR ID Prototype
  - OPFOR Elementary Actions based on Partial BLUEFOR Vulnerability assessment
- **Phase II effort:**
  - Integrate Intel collection planning and action planning with Identification and Vulnerability assessment
  - Integrate with C2 virtual environment (RealWorld)

The Focus of Paper

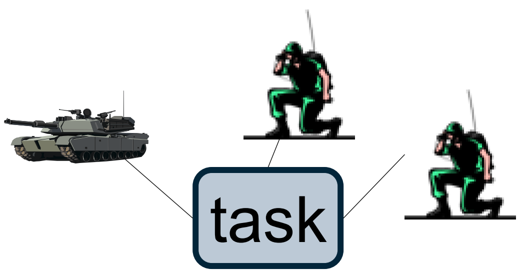
Identification of BLUEFOR operations & plans



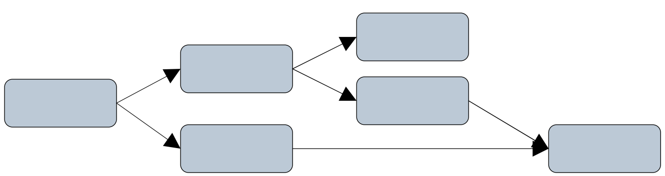
Modeling Details: Quantitative Concepts

- **Asset**
  - An object in the simulation environment
  - Ex: BLUE Recon Squad, Humwee, MP Squad, etc.

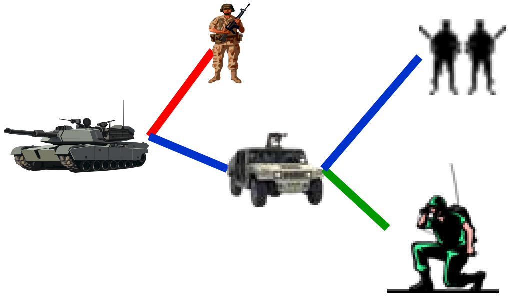
- **Tasks**
  - Coordinated actions that actors want to execute
  - Requires often multiple actions for success
  - Ex: Setup checkpoint, secure site, attack RED positions with fire, etc.



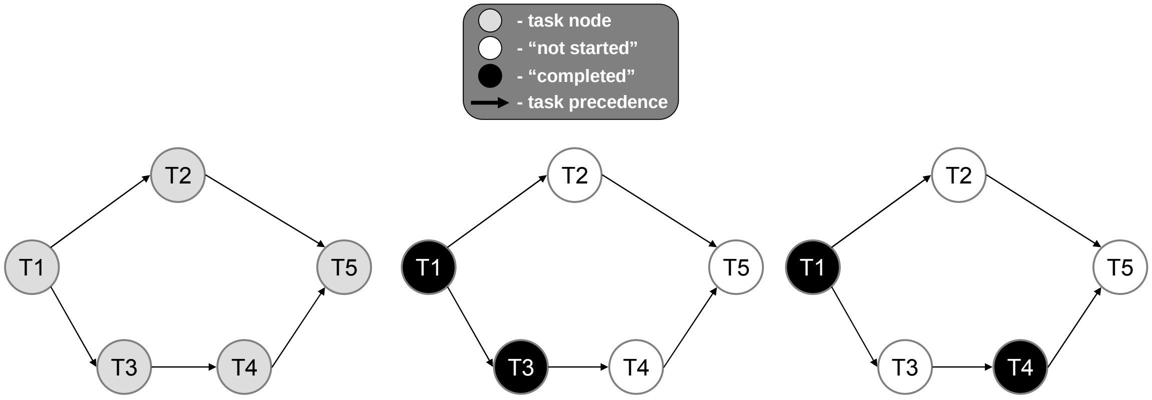
- **Mission Plan**
  - Collection of tasks with precedence
  - Ex: Reconnaissance and patrolling



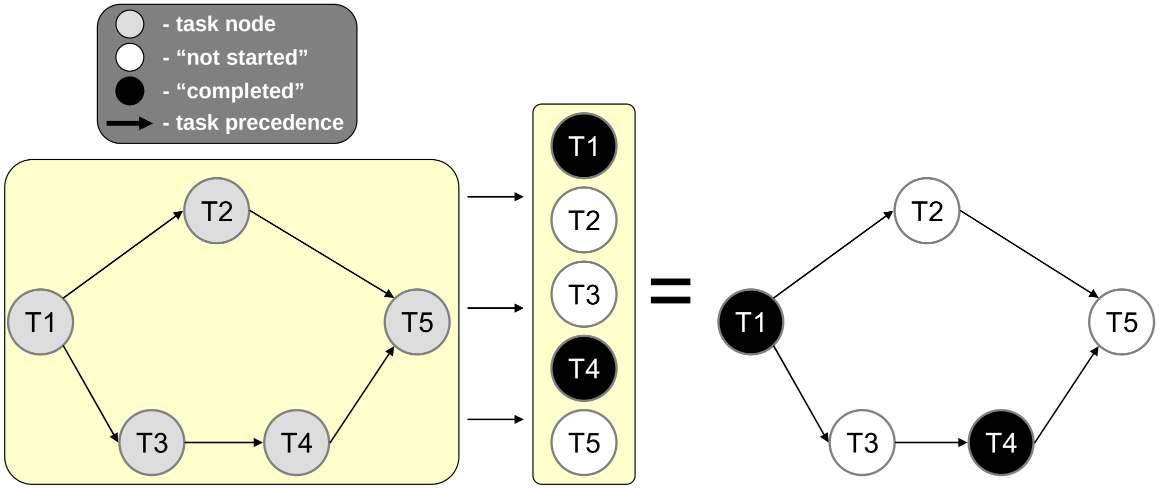
- **Organization**
  - Connections & interactions between actors
  - Ex: a pattern of interactions, actions and meetings of enemy terrorist cell



Mission and Mission State



(a) Mission (b) Feasible Mission State (c) Infeasible Mission State



(a) Mission (b) Observations (c) Observed Mission State

Mission state is defined as **labeled graph**

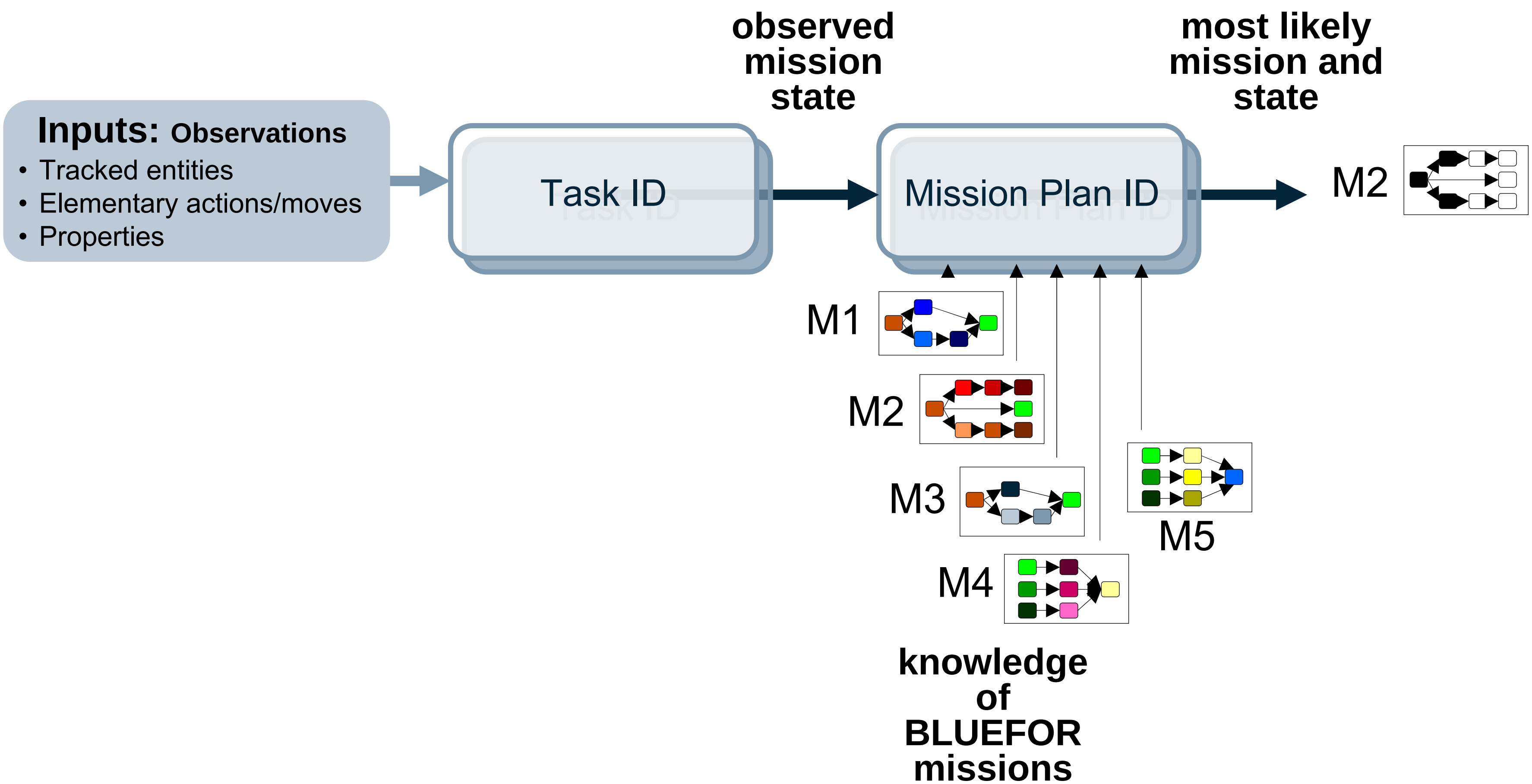
- Labels = states of tasks

Mission states can be **feasible** and **infeasible**

BLUEFOR mission is **partially observable**

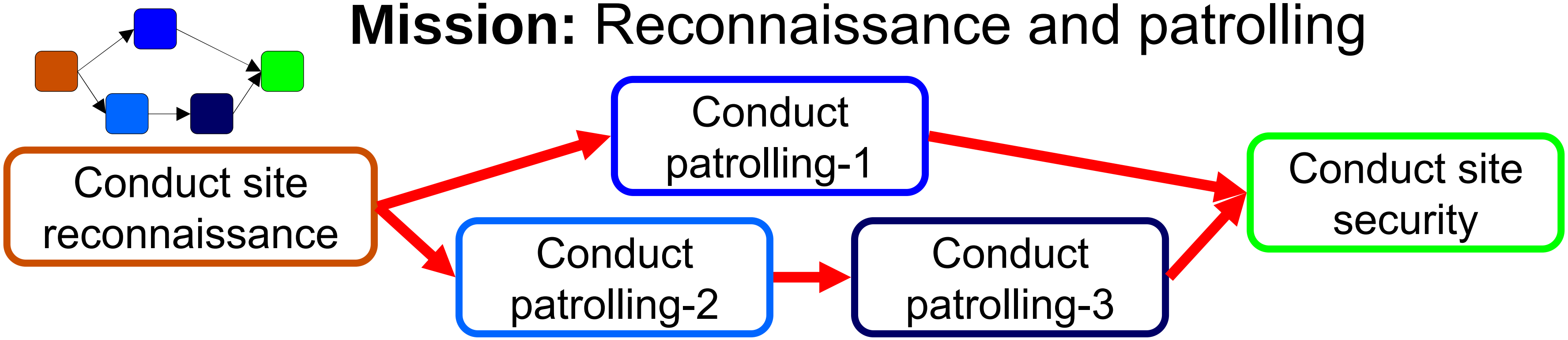
- BLUEFOR mission is hidden from OPFOR
- RED has observation of BLUEFOR’s mission via detected actions
- Observed mission state needs to be **matched** against possible feasible states

Task-Mission Identification



Example of Representing Missions and Tasks

Mission: Reconnaissance and patrolling



Tasks and their requirements

Tasks	SCR	EN	TRSP	MAN	REC	INT	FIRE	DTN	PTRL
Site recon	0	0	0	0	2	0	0	0	0
Site security	3	0	0	0	0	0	0	0	0
Patrolling	0	0	0	0	0	2	0	2	0

Tasks assignment

- Rec Sqd(2)
- Rec Sqd OR RFL Sqd(3)
- MP Sqd(2)

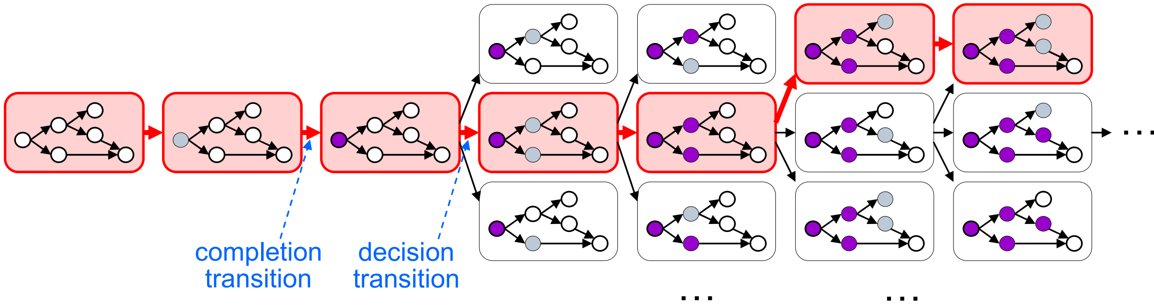
Units and their capabilities

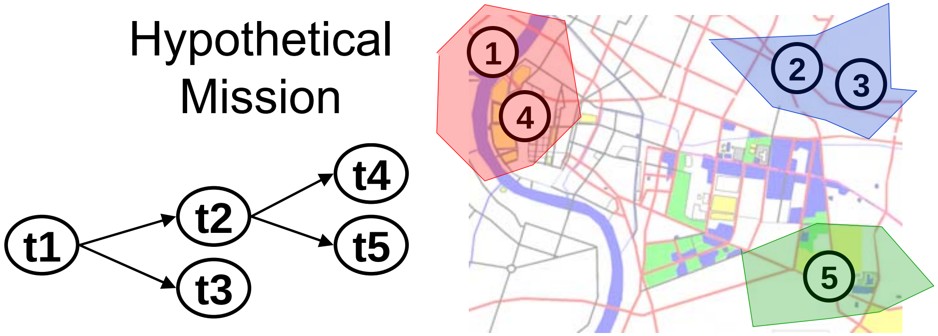
BLUE Units	SCR	EN	TRSP	MAN	REC	INT	FIRE	DTN	PTRL
RFL Squad	1	0	0	1	0	0	0	1	1
Tank	0	1	0	0	0	0	1	0	0
Rec Squad	1	0	0	0	1	0	0	0	0
MP	0	0	0	1	0	1	0	1	1
Humwee	0	0	1	0	0	0	0	0	0

- Each BLUE unit conducts maneuvers and performs the action
- Observation of action are made by OPFOR

Model Property

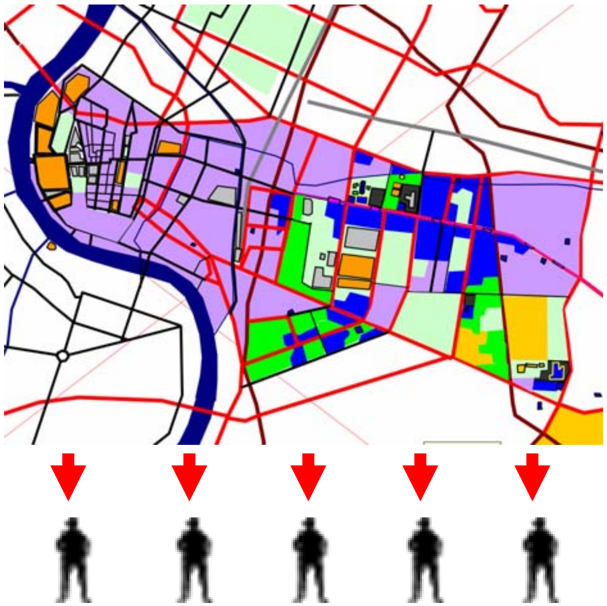
- Allows both temporal and spatial reasoning
- Temporal:
  - Mission state changes over time
    - Actual evolution is limited significantly by precedence and available resources
    - Decision-based transitions – selecting what tasks to do, and what resources to use
- Spatial:
  - Tasks have locations





Example of Observable Data

Environment



RED Sensors

Observations of Activities

1. Geo-spatial information:

- Location of actions and movements

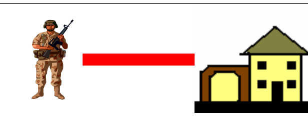
2. Temporal information:

- Time of activities

time

3. Feature information:

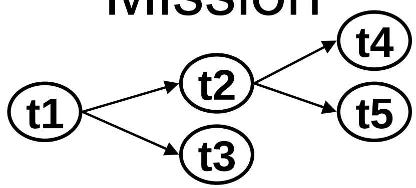
- Who
- Type/features of actors and action (capability applied)

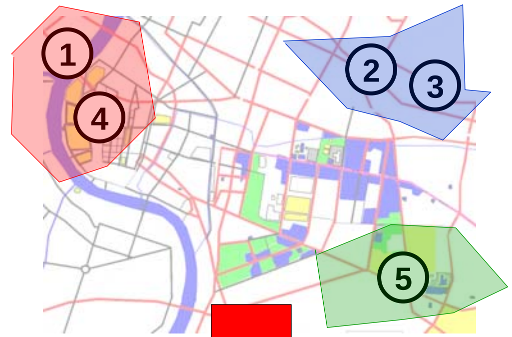


- Action: search
- Duration: 20 min
- Location: Village residential area
- Actors: Rifle Squad

Task ID Solution (1)

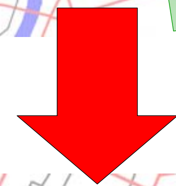
Hypothetical Mission



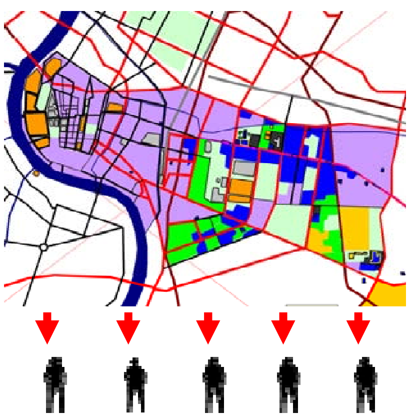


Cluster tasks according to locations

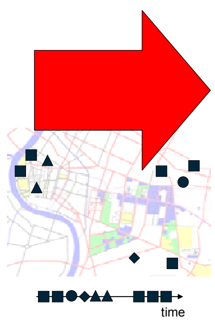
1

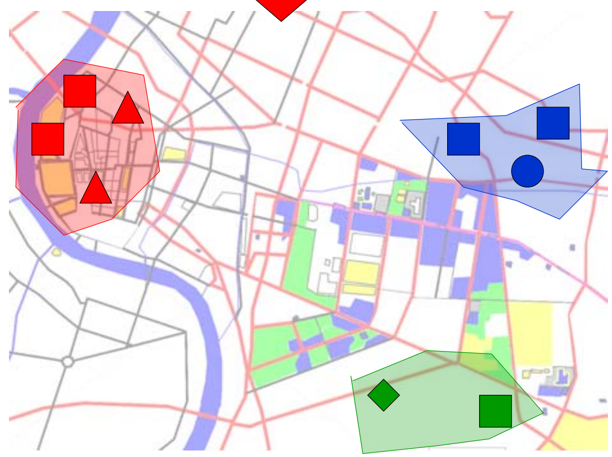


Environment



RED Sensors



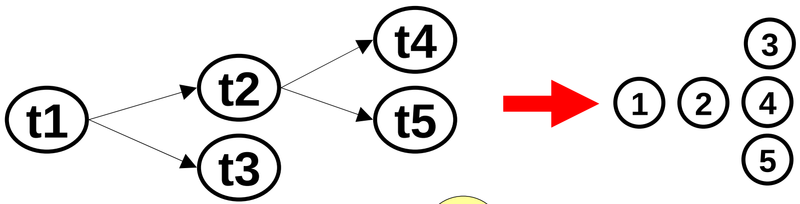


Cluster observations according to areas of tasks

2

Task ID Solution (2)

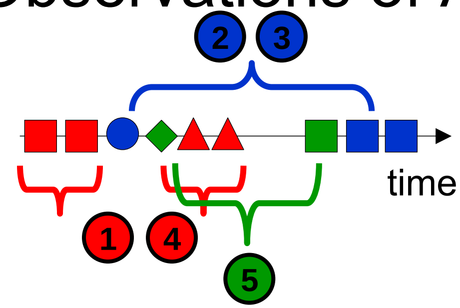
Hypothetical Mission



3a

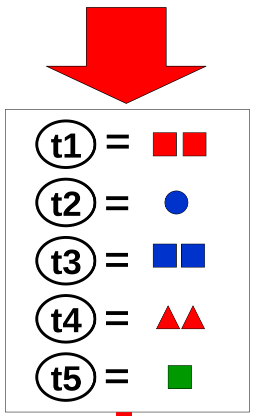
Prioritize tasks according to precedence constraints

Observations of Activities



3b

Cluster observations over time



4

Associate tasks with observations in the order of task priority

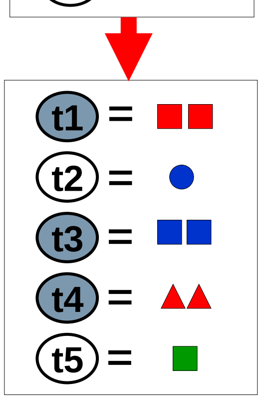
Observed Mission State:

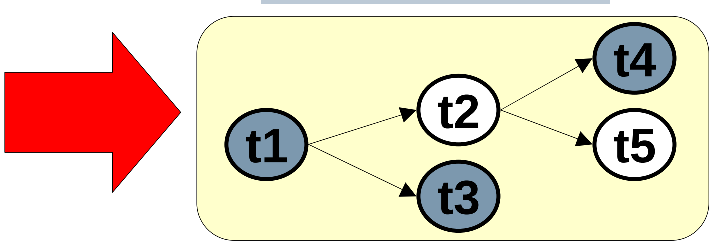
Max Likelihood for Task State Estimates

$$O^* = \max_{O} \prod_{j} p(a_j \mid o_j)$$

5

Perform max-likelihood estimation of task state





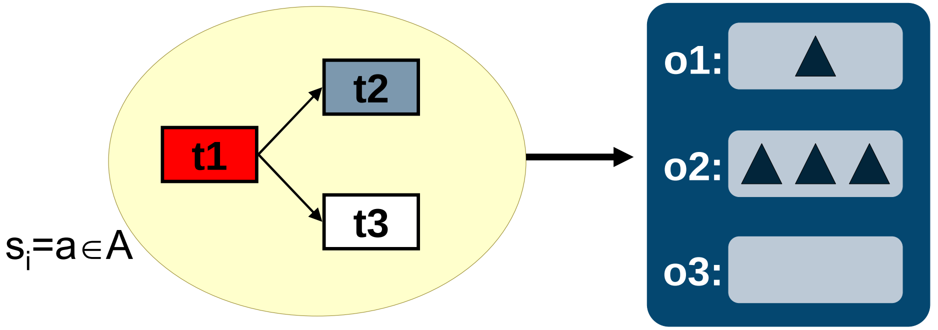
Mission Plan Recognition Model

- **Finding Mission State:** max a-posteriori estimator

$$S^* = \arg\max_S p(S \mid O, m) = \arg\max_S \frac{1}{Z} \prod_i p_i(o_i \mid s_i)\, p_i(s_i \mid s_{N\{i\}})$$
$$\cong \arg\max_S \sum_i \log p_i(o_i \mid s_i) + \sum_i \log p_i(s_i \mid s_{N\{i\}})$$

Hidden Mission State

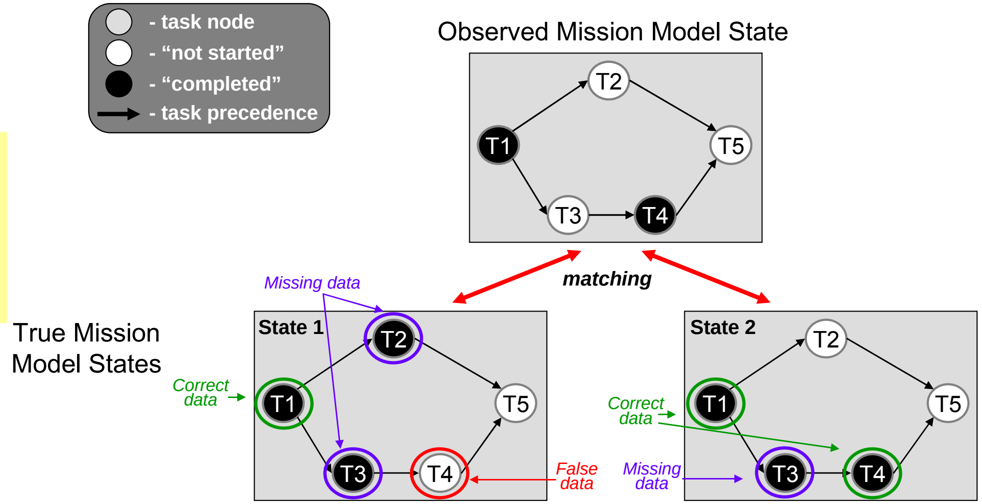
Observations



$P\{s_i \mid s_{N\{i\}}\}$ $P\{o_i \mid s_i\}$

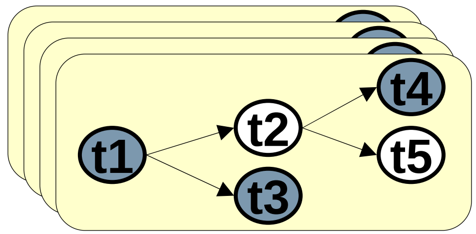
- **Finding Mission:** max likelihood estimator

$$m^* = \arg\max_m p(O \mid m) = \arg\max_m \sum_S p(O \mid S, m)\, p(S \mid m)$$
$$= \arg\max_m \sum_S \prod_i p_i(o_i \mid s_i)\, p_i(s_i \mid s_{N\{i\}})$$



Mission ID Solution

Observed Mission State



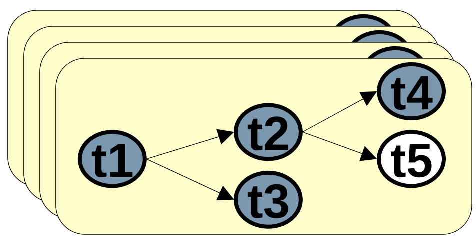
1

Conduct max a-posteriori mission state estimation assuming that task states are uncertain observations

$$S^* = \arg\max_{S} \prod_{i} p(o_i^* \mid s_i)\, p(s_i \mid s_{N\{i\}})$$

Can be solved by dynamic programming, Hidden Markov Random Fields, etc.

Max-posterior Mission States

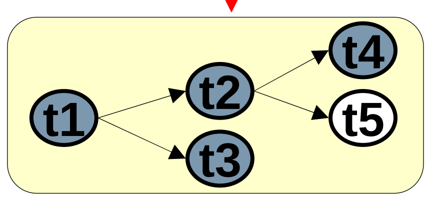


2

Select the mission plan with highest likelihood score

$$\mu(m) = \prod_{j} p(a_j \mid o_j^*) \sum_{S} \prod_{i} p(o_i^* \mid s_i)\, p(s_i \mid s_{N\{i\}})$$

Predicted RED mission & its state

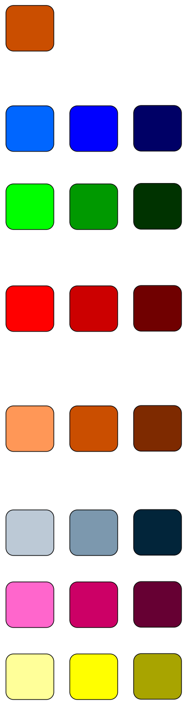


Experiment Summary

- Measures
  – % correct predictions
- Power of the prediction
  – Entropy (inverse of ambiguity)
- Runs
  – For each ground-truth BLUE mission
  – Vary the probability of action/event detection by OPFOR elements
  – Vary time at which predictions are performed

Use-case Details: BLUEFOR Missions

- Tasks (operations)
  - Site reconnaissance
  - Patrolling
  - Site security
  - Attack OPFOR positions
  - Detain OPFOR members
  - Building search
  - Resupply ops
  - Checkpoint setup



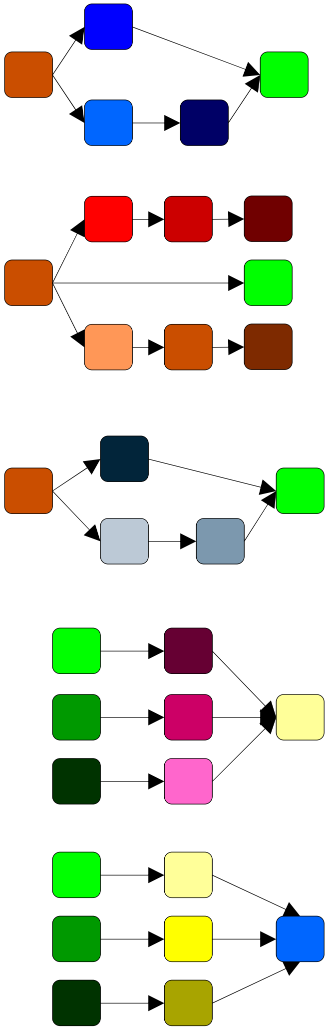
M1: Reconnaissance and patrolling
Site reconnaissance(1)
Patrolling(3)
Site security(1)

M2: Ground Stability and Defensive Ops
Site reconnaissance(1)
Attack OPFOR positions(3)
Detain OPFOR members(3)
Site security(1)

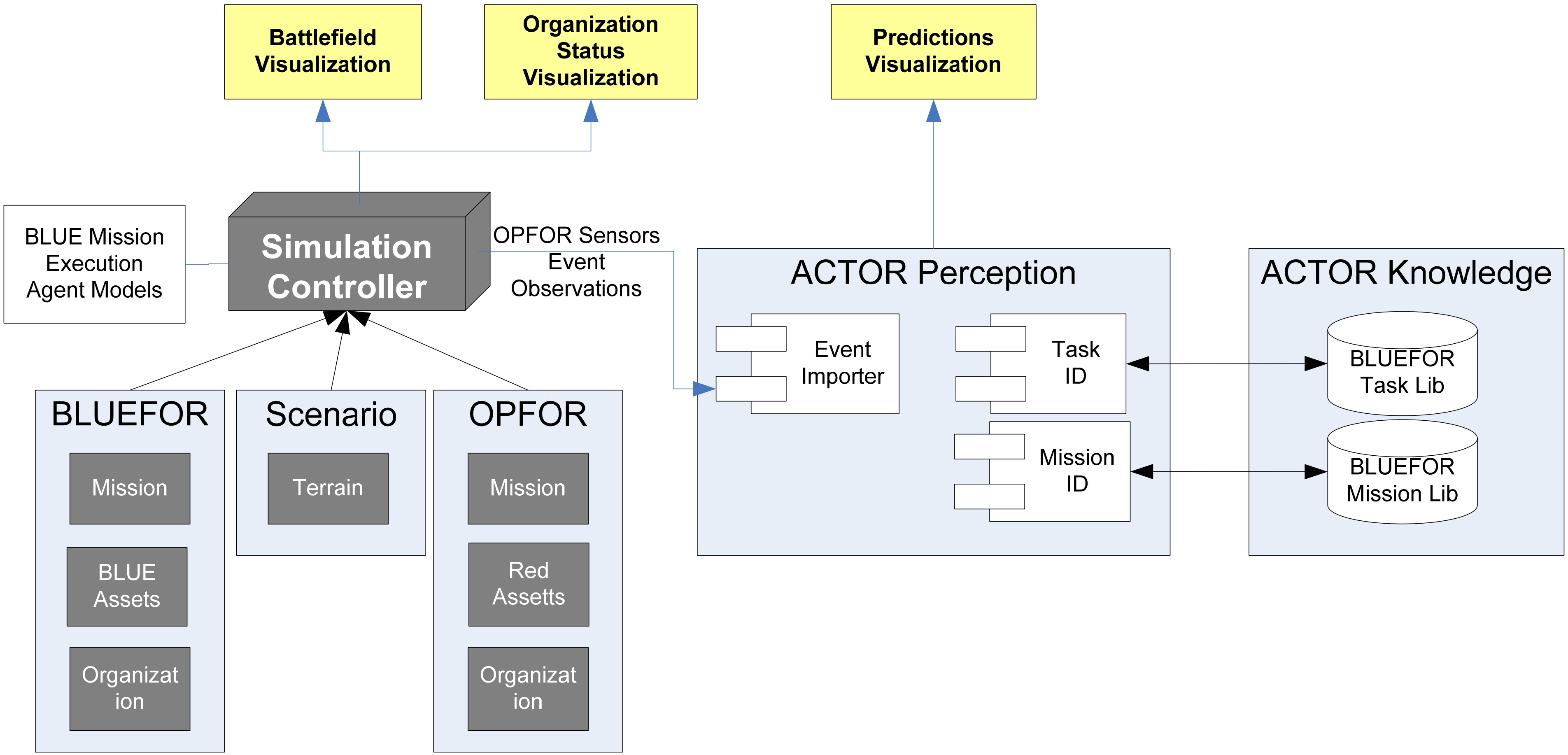
M3: Search
Site reconnaissance(1)
Building search(3)
Site security(1)

M4: Security and supplies
Site security(3)
Resupply ops(3)
Checkpoint setup(1)

M5: Area and site security
Site security(3)
Checkpoint setup(3)
Patrolling(1)

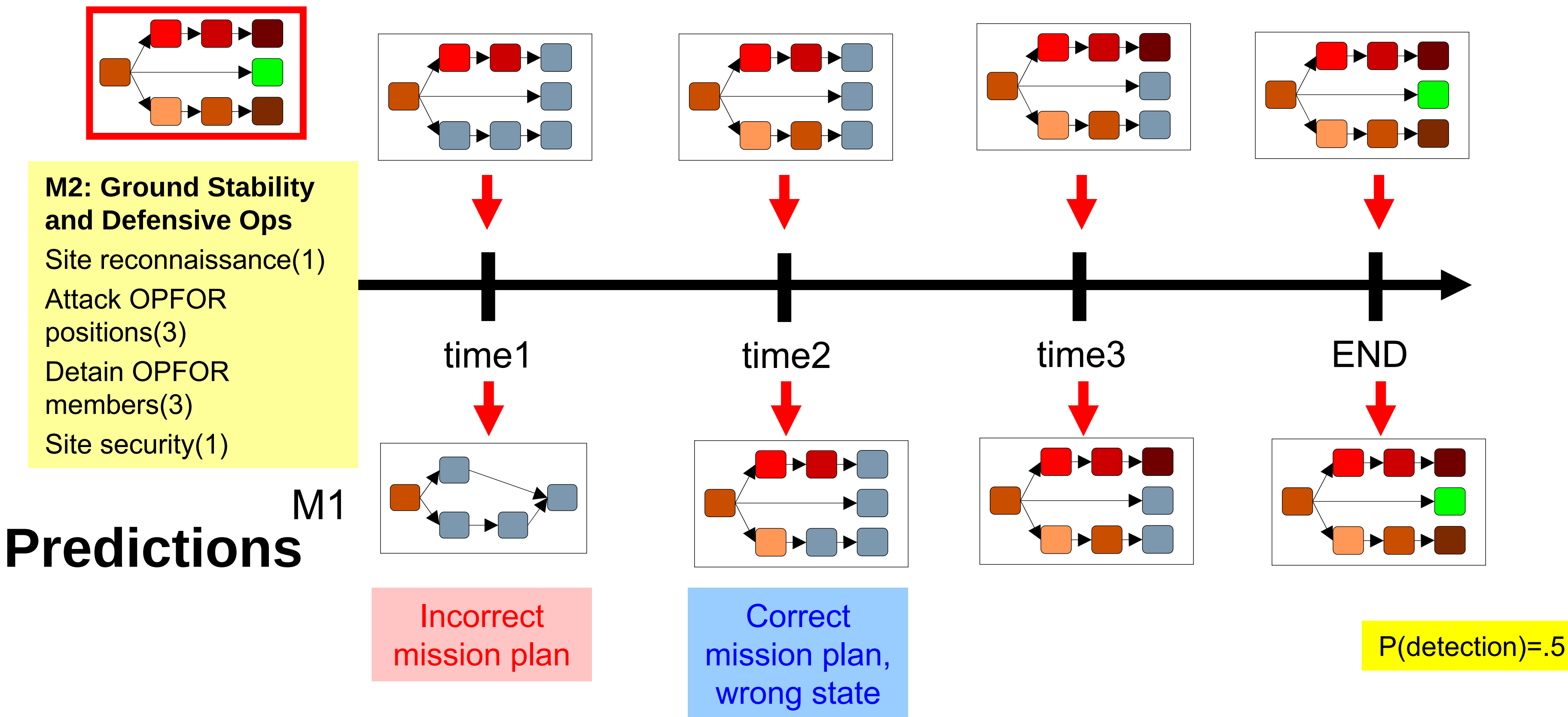


Validation Prototype Architecture



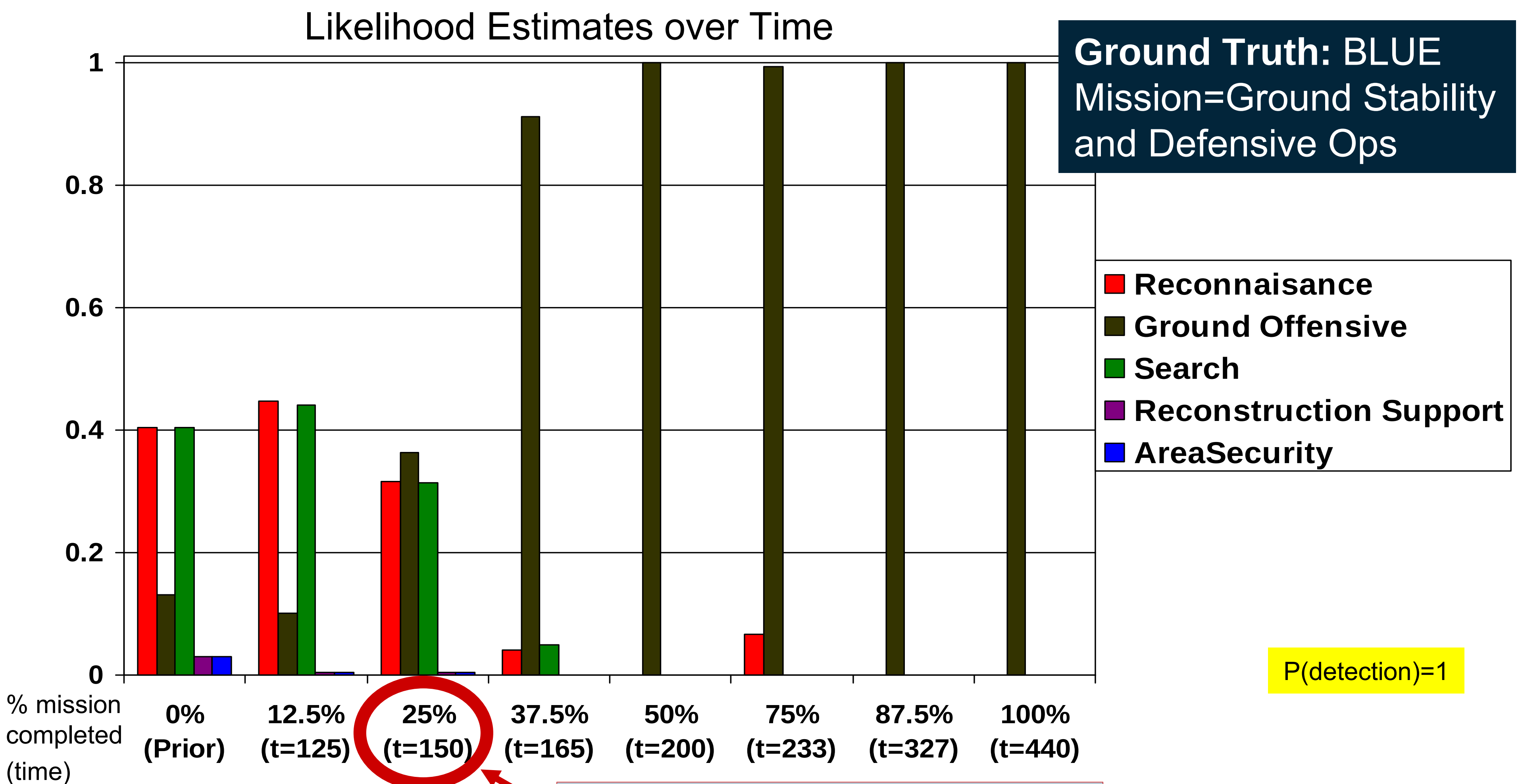
Inference over time-1

Ground Truth



- **Conclusions:**
  – Prediction improves with time as more events are collected

Inference over time-1

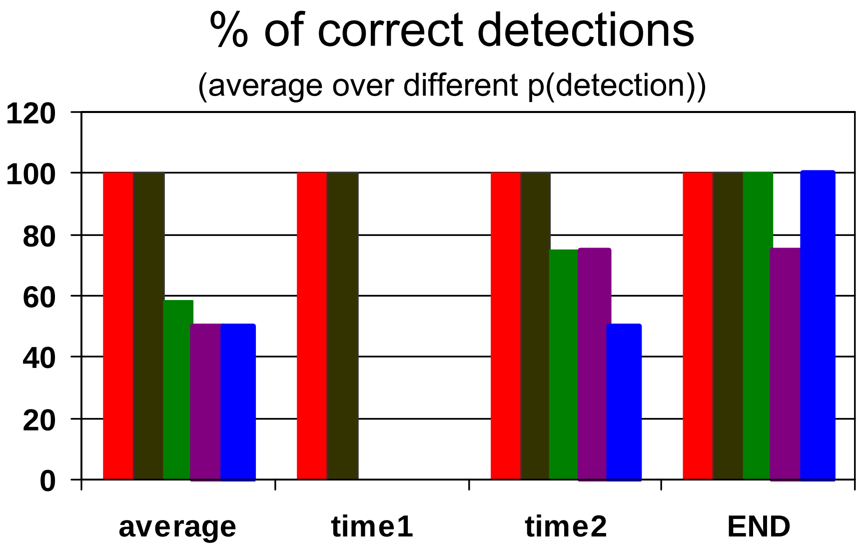


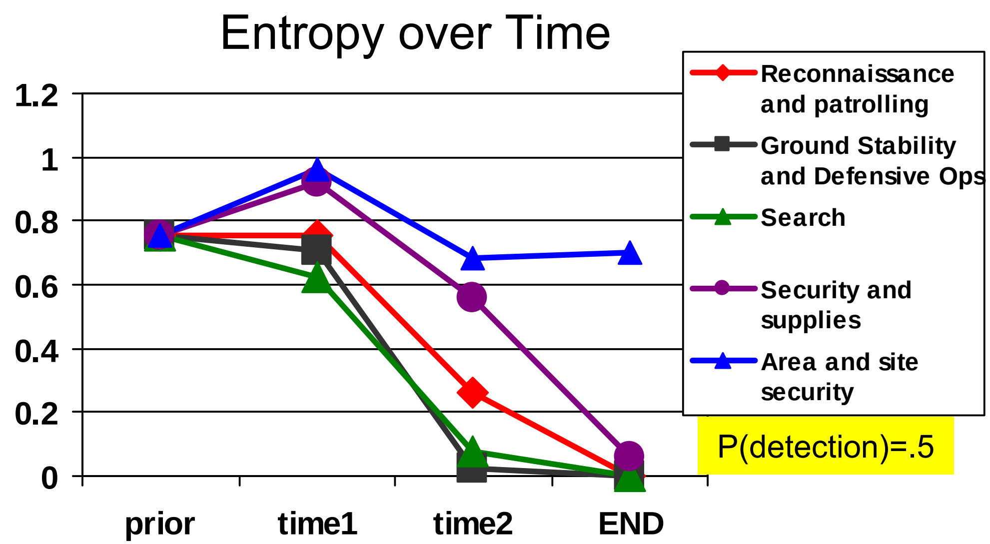
Ground Truth: BLUE Mission=Ground Stability and Defensive Ops

P(detection)=1

Attack (FIRE actions) task completed – start distinguishing hypotheses

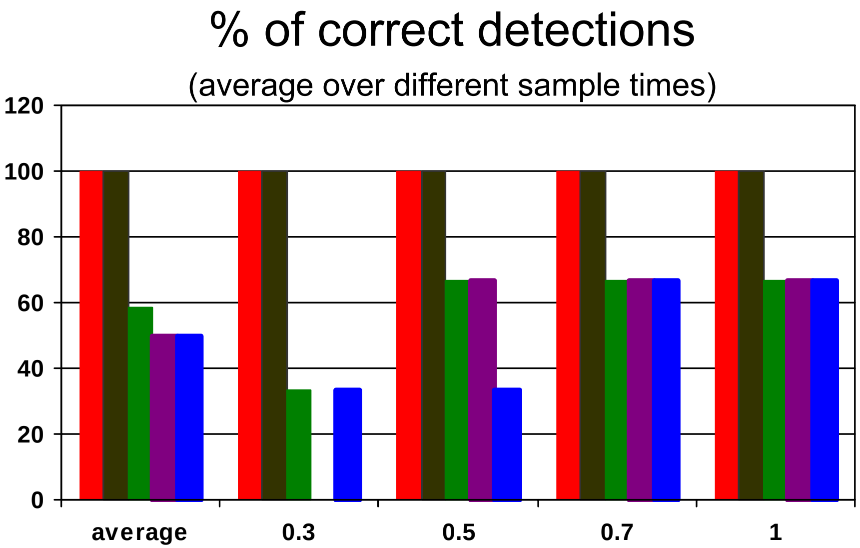
Simulation Results-2: Recognition over time

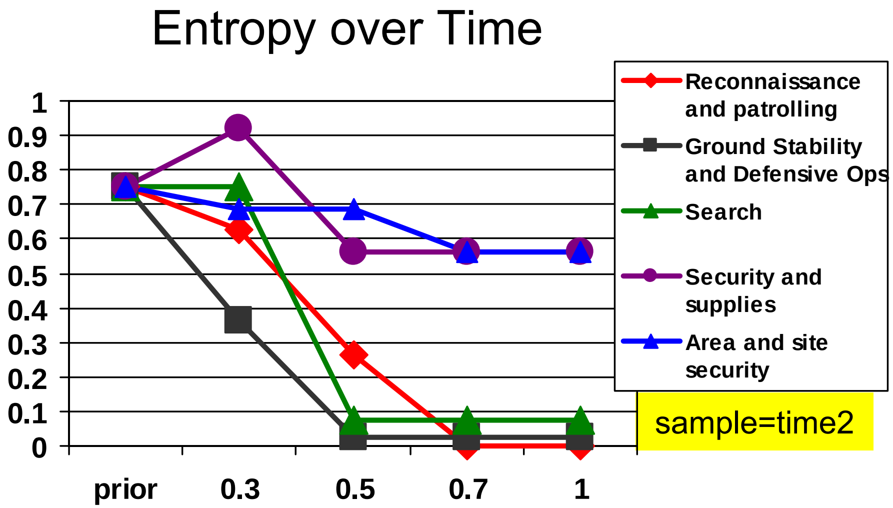




- **Conclusions:**
  - High detection accuracy (>=75%) in the middle of the mission
  - Detection accuracy data supported by entropy (power of estimator)
  - Detection improves towards the end of the mission as more actions are detected

Simulation Results-3: Effect of Detection Probability





- **Conclusions:**
  - Improved accuracy of event detection increases accuracy of predictions
  - Sensitivity to specific mission plan classes is observed

Using Mission Estimates for RED Planning

- Action plans based on org impact:
  - **Effects on organizational resources:** enemy attacks for the *geographic or functional areas* creating resource misutilization and correspondingly a resource shortage at the commander level
  - **Effects on organizational interactions:** events and task requirements that force commanders of BLUEFOR to increasingly *coordinate and synchronize* their activities, resulting in inefficient coordination patterns, loss of information, and correspondingly delayed actions and erroneous decision making by BLUEFOR
  - **Effects on mission/objectives:** vignettes and obstacles that prevent BLUE C2 from *performing specific individual and mission tasks*, thus requiring it to change or abort the mission

Future Plans: Ideas for Phase II

- Automated model building & updating
  - RED ***discovers*** over time the “patterns of BLUEFOR missions/TTPs”
- RED adaptation for plan generation
  - RED ***learns*** over time the impact of its actions and responses of BLUEFOR
- Integration of predictions with OPFOR intelligence collection/sensor planning
  - RED ***designs*** actions to collect data for highest SA